MASTER'S THESIS

Designing a scalable and multi-institutional deployable cardiovascular data intergration solution

De Mulder, W.

Award date:
2021

DESIGNING A SCALABLE AND MULTI-INSTITUTIONAL DEPLOYABLE CARDIOVASCULAR DATA INTEGRATION SOLUTION

by

Werner De Mulder

in partial fulfillment of the requirements for the degree of

Master of Science
in Software Engineering

Student number:		
Course code:	IM9906	
Thesis committee:	dr. ir. Clara Maathuis,	Open University
	dr. Arjen Hommersom,	Open University
External supervisor:	dr. P.H.M. America,	Philips Research

ACKNOWLEDGEMENT

I am incredibly grateful to Philips Research for the opportunity to realize this dissertation concerning an exciting research topic to develop a scalable and multi-institutional software solution to extract cardiovascular information from various clinical data repositories.

I want to express my special thanks of gratitude to my thesis supervisors of the Open University, dr. ir. Clara Maathuis and dr. Arjen Hommersom. It was a privilege to work under their guidance approaching this research. Both taught me to apply the correct research methodology to conduct the research correctly in a scientific way. Within Philips Research, I would like to express my great thanks to my assigned supervisor, dr. Pierre America, fulfilling the role of a principal software architect. Pierre has continuously assisted me and taught me many new things related to interoperability standards in healthcare.

Furthermore, this thesis assignment never came to a successful outcome due to the excellent cooperation with Ziekenhuis Oost-Limburg (ZOL) Genk. ZOL Genk is a hospital with a highly future-oriented technological vision. Under the leadership of cardiologist Dr. Vandervoort and within the Future Health program of ZOL Genk, supporting scientific research, the Future Health program fulfilled an essential role in this research's realization. I understand that this gave me a unique opportunity, for which I am very grateful! I would also like to express my thanks to lic. Bjorn van Dooren, IT Architect Applications of ZOL Genk, for his active contribution to this research. With his broad knowledge as a former EHR developer, Bjorn has taught me a lot of experience about EHR systems and the data management of this software platform.

I fully recognize that accomplishing a master's study, combined with daily work, can never be realized without my loved ones' continuous support. It has not always been easy to align the study with family life, and they contributed enormously to realizing my ultimate dream. For this, I would like to thank my fantastic wife, Annelies Vanderlinden, and my two wonderful sons Torben and Gerben De Mulder, from the bottom of my heart for their support during the three years of study.

Finally, I would like to thank my fellow students Cedric van Labeke and Kurt Stremerch for the excellent cooperation on the assignments, the many discussion moments, and even the nights spent together getting through the master's study.

A big thank you to everyone!
Werner De Mulder

CONTENTS

ABSTRACT

Technological advancements in healthcare are continuously improving the quality of life. Despite all technological innovations, many clinical systems and departments still operate independently, making it challenging to process and exchange the diversity of registered clinical information unambiguously. The recent growth of digital health initiatives and the Covid-19 pandemic of 2020 strengthens the value of data interoperability in healthcare to connect diverse software applications and information systems across different healthcare providers. The lack of data interoperability in various healthcare environments implies that leading healthcare organizations have to put a lot of effort into taking full advantage of the data-intensive culture without losing the meaning of the information, ultimately resulting in loss-making integration projects. This research highlights the need for *standardization* in healthcare.

We aim to propose a scalable software solution applicable in a multi-institutional healthcare environment, reducing implementation burden and improving integration project profitability. The developed solution focuses mainly on the extraction of cardiovascular information from diverse Electronic Health Record (EHR) platforms across Belgian (Flanders) and Dutch healthcare organizations. To avoid losing affinity with healthcare's realistic complexity, we designed a software architecture and a prototype in collaboration with a Belgian medical institution acting as a vendor-specific EHR reference site. The proposed extraction method relies on a multi-institutional extraction method using vendor-specific standardized clinical content. Subsequently, we applied an efficient transformation process on the returned standardized dataset to deliver an unambiguously defined clinical dataset.

The proposed solution's evaluation relies on three different pillars whereby we validated the proposed solution on correctness, efficiency, and performance. The applied validation mechanism results in an accurate software solution that meets the demands supporting the cardiovascular workflow. The efficiency evaluation methodology relies on a cost model estimating the implementation cost required to implement the prototype in another medical setting and estimates an expected cost to scale up the prototype accommodating additional clinical concepts. Although the validation results do not fully reflect reality due to restrictive factors we had to consider during the implementation phase, the prototype satisfies the demand to offer an accurate, efficient, and performing solution for the unambiguous representation of cardiovascular data into a multi-institutional setting.

Future research can contribute to scale up the proposed software solution to extract a broader range of cardiovascular concepts. Since the proposed solution focuses only on one EHR vendor, further investigation into similar multi-vendor EHR extraction methods could make the software solution broadly deployable. Moreover, additional research can give us better insights into semantic data mappings' maintainability if we use dedicated terminology solutions in the software architecture.

1. INTRODUCTION

In recent years we have seen a massive increase in the amount of generated data [Botta et al., 2016]. A similar trend applies within the healthcare industry due to rapid digitization and new image acquisition technologies, producing large quantities of data through the continuum of care [Belle et al., 2015]. Healthcare providers collect clinical data in various domains, historically obtained during patient care and hospital admissions. Several clinical resources contain specific data holding a valuable source of clinical information, and each constitutes fundamental elements of insight that support medical decision-making [Ashfaq and Nowaczyk, 2019]. Consequently, multi-source data collection is a critical aspect of ensuring clinical decision-making validity. In particular, data collection is a vital aspect in preparing cardiovascular procedures, in which cardiologists diagnose and treat certain cardiovascular conditions of cardiac disease patients.

Since the required data is spread across various autonomous clinical data sources, manually collecting such information is challenging and time-consuming. Additionally, significant data diversity exists among many multi-vendor clinical source systems implemented by medical providers leaving clinical practitioners at a disadvantage to harness the value of siloed clinical information. As a result, healthcare providers have to deliver a lot of effort to collect cardiovascular data in an interpretable manner from multiple clinical data sources. Unfortunately, the diversity between various clinical settings implies that healthcare providers have to repeat these efforts to obtain a comparable data set. Consequently, healthcare providers encounter high integration costs, making integration projects loss-making. As a central research question, we investigate how to design a scalable and multi-institutionally deployable software solution to collect cardiovascular clinical data from multiple data sources in a standardized manner while preserving the meaning and context of the collected data interpretable by both humans and machines. The solution aims to reduce the implementation burden and the resulting high implementation costs that healthcare organizations encounter to obtain cardiovascular data across Belgian (Flemish) and Dutch healthcare providers. To align the proposed solution with practice, we develop a prototype in a Belgian hospital focusing on developing and applying clinical care innovation. The knowledge obtained from the practical implementation constitutes the basis to deliver a valuable solution and contributes to estimating the proposed solution's efficiency.

This research is divided into several sections. Section 2 presents a literature study of related work containing crucial concepts that we consider in the research. The related work section emphasizes the generic concept of interoperability and introduces several healthcare interoperability standards. Further, it investigates data integration methods to retain the clinical meaning of the collected information. This section also covers a literature study about the use of a clinical data model, accommodating the continuously changing environment across different healthcare institutions. We conclude this section with a short discussion to justify the decisions taken based on the investigated literature. In section 3, we describe the complete cardiovascular data documentation process and indicate precisely the positioning of the discussed use cases within this process. Section 4 deals with the research methodology and technical aspect of this research and presents all sub-research questions in more detail, breaking the central research question down into several challenges. This chapter also gives a detailed overview of the scientific research method used to explain how we want to approach the problem and find a solution [Peffers et al., 2007]. After analyzing the diversity present between various clinical repositories, we investigate in

section 5 the EHR market share across Belgian and Dutch healthcare organizations to gain a better picture of the EHR landscape in this region. Based on the outcome of the market analysis, we describe in section 6 design decisions to satisfy the proposed solution's multi-institutional perspective. In addition to these facts, section 6 also discusses how to accommodate possible variations between various ChipSoft EHR implementations due to locally implemented customization decisions to deliver a uniform dataset aligned with our software solution. We further elaborate on achieving a syntactic and semantically harmonized representation of the source dataset to comply with a target clinical data model. Section 7 describes the entire software prototype's implementation, employing the previous section's research design decisions. This section aims to align our prototype software solution with a real clinical environment to reveal shortcomings in a clinical setting we overlooked earlier during the research design. The following section 8 deals with the validation of the proposed solution and discusses the validation process to verify the correctness of the generated clinical datasets and compliance with the applied data model. This section also explains the applied cost model reporting the proposed solution's efficiency estimating the amount of time required to implement the prototype in other clinical settings or build a more comprehensive cardiovascular dataset. In section 9 we summarize the results to all sub-research questions answering the central research question. We conclude this study in section 10, where we briefly discuss potential future research perspectives aligned with the current investigation.

2. Related Work

Patients' healthcare data originate from many different source systems captured all across various medical departments. The aggregation of clinical data into a central repository of information is a crucial process to deliver physicians a valuable dataset representing patient-related key insights from previous studies. In many cases, the diverse data representation of clinical data is not uniform across all subsystems. Consequently, data diversity and especially the interpretation of the clinical context are challenging in healthcare [Bhartiya and Mehrotra, 2014]. As the number of data science applications in healthcare raises significantly, we must deliver maximum efforts to present ambiguous medical information in an unambiguous and standardized way to make clinical data interpretable and interchangeable [Waring et al., 2020]. Even in the face of the current Covid-19 pandemic, we have never had a greater demand to share clinical information between diverse software solutions spread across the globe [O'Reilly-Shah et al., 2020]. Consequently, it is potentially significant that data interoperability challenges do not prevent the exchange of clinical data between various data solutions, making data interoperability a core concept in healthcare. The Data Interoperability Standards Consortium[1] defines interoperability as *'the ability of systems and services that create, exchange and consume data to have clear, shared expectations for the contents, context and meaning of that data'*. In subsection 2.1, we define interoperability in healthcare and subsequently reflect why this concept is essential to maximize digitization efficiency. This subsection also briefly discusses two crucial interoperability concepts that significantly contribute to the unambiguous communication of clinical data [Lehne et al., 2019]. A first concept handles the *syntactic interoperability*, presenting several formats to exchange clinical information. In contrast, the second

[1] https://datainteroperability.org/

concept emphasizes the *semantic interoperability*, dealing with exchanging clinical data employing a common vocabulary, enabling receiving systems to eliminate unambiguity to promote accurate and reliable communication. In subsection 2.2, we investigate literature on diverse methodologies discussing *how to extract* clinical data, taking into account existing data variations originating from various clinical data repositories (CDRs). We conducted in subsection 2.3 a similar literature study focusing on the reusability of clinical data in a multi-institutional environment and further discuss and evaluate a *reference data model* to increase the level of interoperability spread across various healthcare institutions. In section 2.4 we give some background information on the integration engine to deliver a common data structure against multi-vendor CDRs. To conclude, we argue in subsection 2.5 the conclusions taken based on earlier reviewed literature outlining the foundation of our study.

2.1. The role of standards in healthcare IT

Interoperability is in many businesses of significant value, helping enterprises automate data processing [McMillan et al., 2017]. Easy access to data and seamless sharing of information between different software applications is a prerequisite empowering digitalization to foster collaboration. In addition to being essential in various industries, data interoperability is also crucial in healthcare to enable the seamless exchange of health data between different healthcare systems and health organizations to facilitate the exchange and interpretation of medical data [Lehne et al., 2019]. In the context of healthcare, interoperability facilitates seamless exchange of health data between different healthcare systems and organizations with the ability to interpret and use the exchanged data consistently. The collection of incorrect data can impact the decisions taken by the cardiologist with catastrophic consequences for the patient. To achieve a satisfactory level of interoperability, we want to elaborate further on both concepts of interoperability defined by Lehne et al. [2019]. We will discuss first some standards to align the exchange of data between different healthcare systems, ensuring the correct processing of the message content. Second, we elaborate on how to maintain the context and meaning of the exchanged clinical data since the use of an exchange standard is not sufficient to ensure a correct interpretation of the transferred messages [Adel et al., 2018].

Syntactic interoperability

Syntactic interoperability ensures the ability to exchange information between systems according to a predefined data format [Lehne et al., 2019]. Today, healthcare organizations apply healthcare messaging standards to exchange data in a structured style between different clinical data sources, promoting integration and sharing across independent source systems [Dogac et al., 2007]. Health Level Seven (HL7[2]) version 2 (v2) is the primary data format standard to exchange clinical data, represented as a delimited file format containing embedded information without passing attribute information along with it. In figure 1, we depict an example of the HL7 v2 message representation. HL7 v2 messages are created based on a trigger event associated with a specific message type. Figure 1 represents an HL7 message triggered by a patient registration (A04 message type colored in yellow) into the

[2]https://www.hl7.org/

clinical environment to notify another application a specific patient has arrived. An HL7 v2 message consists of a hierarchical message structure containing various segments, represented as a single line. For example, in figure 1 we represented two segments red-colored to keep a clear overview of the complete HL7 message structure. Each HL7 segment begins with a header (green-colored) and defines a logical group of data fields, identified with a three-character coded segment identifier. The first message header (MSH) always determines the starting point of the HL7 message. Other message segments, such as the PID segment, comprise patient demographics data, while the PV1 segment focuses more on the visit information. Furthermore, each message segment consists of various fields separated by pipe characters. In some cases, specific HL7 fields can contain multiple values composed of multiple components separated by a carat sign, as shown in blue. Those components could consequently contain various subcomponents to represent more complex data structures. The lack of robust agreements about the order, content, and semantics of the data makes integration complex and time-consuming, causing software vendors to implement the HL7 standards differently [Beeler, 1998]. This substantial diversity in HL7 implementation makes data integration costly by slowing down integration cycles and making them difficult to manage.

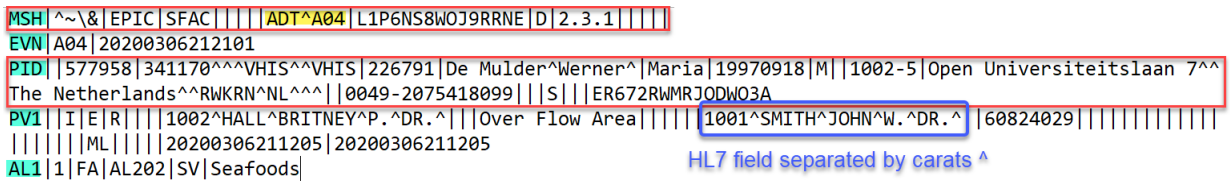

Figure 1: HL7 version 2 example message

As a result of a broader implementation of the HL7 v2 standard in various clinical domains, the demand increased to develop a standard that supports more complex, domain-specific, and reusable message formats for the event-oriented workflow. The lack of an existing information model within the HL7 v2 standard resulted in the development of a new HL7 version 3 (v3) standard, based on a reference information model (RIM) [Beeler, 1998]. The RIM defines an object-oriented approach leading to a fully specified, robust standard to represent clinical data entities (e.g., patient administration, laboratory observation), fostering interoperability. Based on the RIM, the HL7 Clinical Document Architecture (CDA) has been developed to support the exchange of clinical documents using the XML[3] standard. Figure 2 depicts a partial HL7 v3 message example in which the XML document clearly outlines the body of the message. We emphasized the object-oriented structure by highlighting the patient-oriented object in yellow and the result-oriented object in red. An object identifier uniquely identifies each resource to ensure global uniqueness (green colored).

The flexibility, and consequently the broad implementation (see figure 3) of the HL7 v2 standard, combined with the lack of backwards compatibility of HL7 v3, resulted in far too high implementation costs restricting the transition towards HL7 v3 in healthcare [Bender and Sartipi, 2013]. Consequently, as depicted in figure 3, HL7 v2 is today the most widely

[3]https://www.w3.org/XML/

```
<recordTarget>
  <patientClinical>
    <id root="2.16.840.1.113883.19.1122.5" extension="444-22-2222" assigningAuthorityName="UZ Patient ID"/>
    <statusCode code="active"/>
    <patientPerson>
      <name use="L">
        <given>Werner</given>
        <given></given>
        <family>De Mulder</family>
      </name>
      <asOtherIDs>
        <id extension="BSN 123456" assigningAuthorityName="SSN" root="2.16.840.1.113883.4.1"/>
      </asOtherIDs>
    </patientPerson>
  </patientClinical>
</recordTarget>
<observationEvent>
  <id root="2.16.840.1.113883.19.1122.4" extension="1045813" assigningAuthorityName="UZ LAB Filler Orders"/>
  <code code="1554-5" codeSystemName="LN" codeSystem="2.16.840.1.113883.6.1" displayName="GLUCOSE^POST 12H CFST:MCNC:PT:SER/PLAS:QN"/>
  <statusCode code="completed"/>
  <effectiveTime value="202102251545"/>
  <priorityCode code="R"/>
  <confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25"/>
  <value xsi:type="PQ" value="165" unit="mg/dL"/>
  <interpretationCode code="H"/>
  <referenceRange>
    <interpretationRange>
      <value xsi:type="IVL_PQ">
        <low value="70" unit="mg/dL"/>
        <high value="105" unit="mg/dL"/>
      </value>
      <interpretationCode code="N"/>
    </interpretationRange>
  </referenceRange>
</observationEvent>
```

Figure 2: HL7 version 3 partial example message

used standard. The internal version numbers represented on this figure refer to the continuous updates of the standard that ensure continuity with current healthcare requirements.

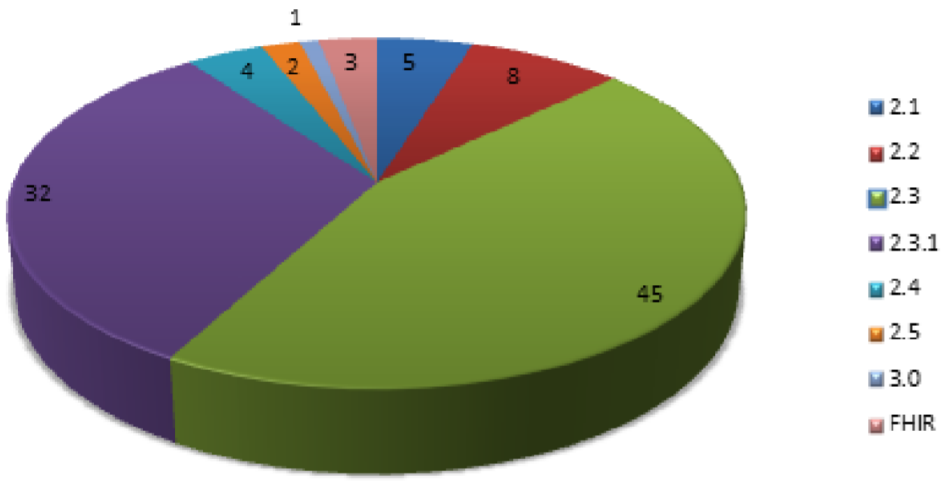

Figure 3: Industrial usage of HL7 standards [Joyia et al., 2018]

Data exchange in HL7 v2 and v3 typically flows according to a point-to-point communication workflow, represented in figure 4. Data communication is managed by an HL7 integration engine, positioned in the middle between the various clinical information system interfaces and the Electronic Health Record (EHR). The International Organization for

Standardization (ISO) defines an EHR as *a repository of information about the health status of a subject of concern, in computer-processable form* [Kwak, 2005]. The amount of information contained by an EHR is highly dependent on the extent to which different information systems are integrated and varies strongly per medical institution. The HL7 integration engine provides flow control, data transformation and central monitoring between different applications from various clinical departments.

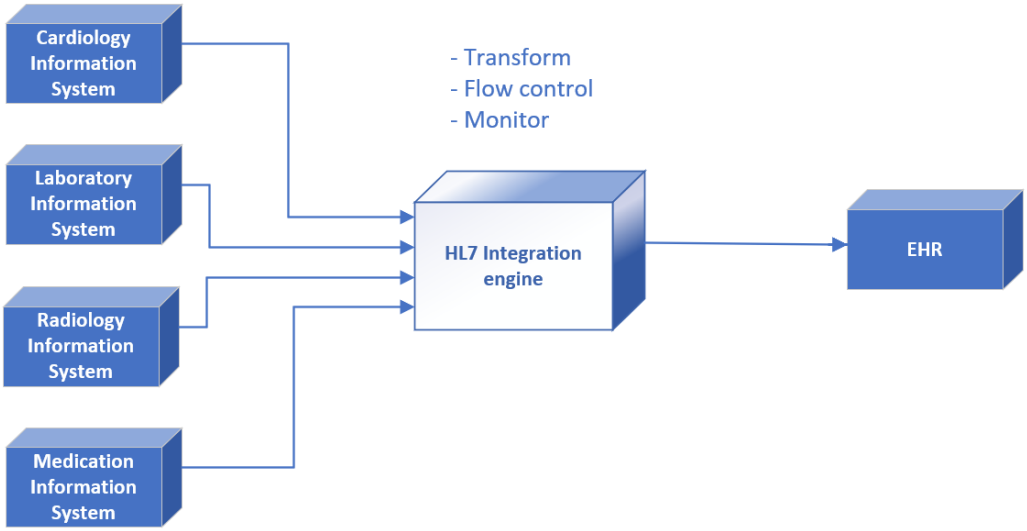

Figure 4: HL7 data flow example managed by an integration engine

Today, Fast Healthcare Interoperability Resources (FHIR[4]) is the newest HL7 standard for digitally exchanging healthcare data, addressing shortcomings in the previous HL7 standards. FHIR uses open standards and supports existing health and Internet Standards such as HTTP, XML and JSON[5]. The standard has a resource-oriented architecture providing atomic data access and supports a RESTful (REpresentational State Transfer) architecture to manage resources [Fielding and Taylor, 2000]. FHIR is built around a REST API model which is a better-formalized approach to exchange clinical data and provides a more efficient way to query and retrieve clinical data [Ismail et al., 2016].

FHIR differentiates from previous HL7 standards as it is possible to make a specific request (e.g., a heart rate observation) without transmitting the complete message and parsing the content. The *atomic data access* architecture is a considerable advantage compared to the HL7 v2 and v3 standards. Another benefit of the FHIR standard is related to the ability allowing clinical data providers examining the context of the obtained data using web APIs. Web APIs offer a lightweight method to access and store clinical data in FHIR-compatible databases accessible to clinicians and researchers for analysis. FHIR provides software developers with the ability to create new applications by fully leveraging the underlying healthcare IT system and introduce opportunities to grant patients secure access to health data from mobile apps outside the hospital [Mandel et al., 2016]. FHIR is a continuously evolving standard Benson [2010] following an Agile development process Manifesto

[4]http://hl7.org/fhir/
[5]https://www.json.org/

[2001]. Each FHIR specification[6] is referencing to a maturity level describing the extent to which the resources within this standard are mature. Maturity levels vary from level zero (draft) to level six (normative). Standards for Trial Use (STU) specifications allow developers to practice the specification in real-world implementations before the specification enters a normative version. Today, the most relevant specifications are the second draft standard for trial use (DSTU2), the third standard for trial use (STU3), and FHIR release 4 (R4). The STU3 standard, published in 2017, is a full FHIR release for trial use where none of the resources are normative. FHIR Release 4 is a new FHIR release, released in 2019, containing changes and interoperability enhancements to promote interoperability in exchanging health data between healthcare institutions. The more current FHIR release 4 is a standard that has been tested very intensively in practice and has gone through the full development cycle of the FHIR maturity model. Consequently, this state-of-the-art FHIR standard includes normative content that should enable developers to apply the standard consistently and universally. The normative nature of an FHIR standard ensures that an FHIR resource's structure cannot change.

SEMANTIC INTEROPERABILITY

While syntactic interoperability describes the clinical dataset structure, semantic interoperability focuses on the messages' content [Lehne et al., 2019]. Because health data exists in many forms (e.g., laboratory results, vital signs, clinical documents), the diversity of clinical data spanning various CDRs results in interoperability challenges to interpret the meaning and context of different clinical concepts consistently [Bhartiya and Mehrotra, 2014]. Focusing on collecting clinical data, we can mainly categorize health data into two forms, structured and unstructured data. Many literature reviews address the diversities between unstructured and structured data in healthcare [Kruse et al., 2016] [Fong et al., 2015]. Structured data is represented as consistent and organized clinical data, while unstructured data lacks this organization and corresponds with unorganized and irregular datasets introducing ambiguity, making analysis much more challenging. Parsing unstructured healthcare data is associated with the textual processing of medical reports to fetch selective medical conclusions. Today, an investigation is already ongoing within Philips Research to find accurate solutions for processing unstructured data. Consequently, we focus this research only on structured data residing in predefined data fields.

Still, the lack of unambiguously defined clinical data concepts challenges healthcare providers and organizations to ensure efficient care. International terminology standards offer a solution to control locally applied health vocabularies by identifying specific observations consistently and unambiguously. Logical Observation Identifiers Names and Codes (LOINC[7]) is the international standard for identifying clinical and laboratory observations. A LOINC code represents a unique code for a particular observation. On the other hand, Systemized Nomenclature of Medicine Clinical Terms (SNOMED-CT[8]) is a multilingual international terminology standard applied to encode clinical data supporting semantic interoperability. Figure 5 refers to a specific segment of an HL7 v2 message containing a LOINC code to identify and request an observation. The LOINC code '600-7', refers to the

[6] https://www.hl7.org/fhir/versions.html
[7] https://loinc.org/
[8] https://www.snomed.org/

common name 'Bacteria identified in Blood by Culture'. The orange text box indicates the request for the observation. The green text box points to the corresponding demand's result containing the SNOMED-CT identifier and name, providing an unambiguous unique reference to the clinical concept.

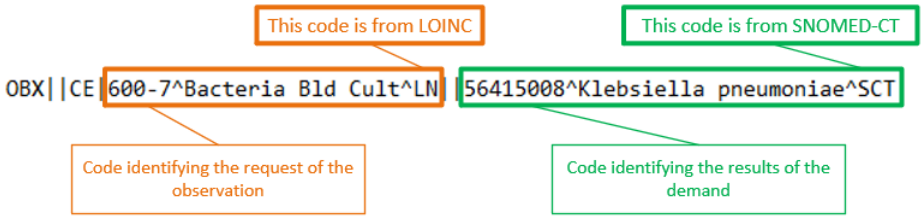

Figure 5: Example of HL7 message containing a codified test result

Recent developments address the variety present between different terminologies to define particular clinical concepts [Saripalle et al., 2020]. Unified Medical Language System (UMLS[9]) describes a terminology standard addressing the existing diversity among disparate clinical terminology standards. Through UMLS, we can design an abstract layer on top of diverse terminology standards by applying a unified global terminology. The international character of this research and the desire to find a solution that unambiguously represents clinical concepts from various CDRs triggers us to bring several semantic standards together and represent them as one single entity. The representation of differently expressed similar clinical concepts as one entity and the ability to maintain the relationships among various terminology standards can enable third-party software applications to represent the data in any terminology recognized by the UMLS standard. Additionally, the UMLS standard can assist in finding the corresponding entity defined by diverse terminology standards. In figure 6 we briefly explain the UMLS concept based on a practical example. For example, if we consider the concept of stress echocardiography, we see that this concept is defined slightly differently within different terminology standards but having the same meaning. The terminology standards layer in figure 6 represents this existing diversity among diverse terminologies. These clinical concepts refer within the UMLS standard to Atom Unique Identifiers (AUI), depicted as the middle layer in figure 6. The orange highlighted text exposes slightly different names assigned to each clinical concept. For example, we indicated three slightly different Dutch expressions holding the same meaning. Subsequently, the atoms have a relationship with a Concept Unique Identifier (CUI), expressing all underlying terminology concepts as one single entity.

Despite international terminology standards that can significantly improve data quality by unambiguously representing clinical concepts, it remains practically a considerable challenge to employ these standards in practice. Metke-Jimenez et al. [2018] designed a solution to access diverse clinical terminology standards hiding the underlying software solution's complexity. The solution, called Ontoserver, provides unified access to multiple clinical vocabularies and facilitates a mechanism to keep various chained terminology solutions up-to-date with the most actual clinical terminologies. Consequently, the discussed

[9]https://www.nlm.nih.gov/research/umls

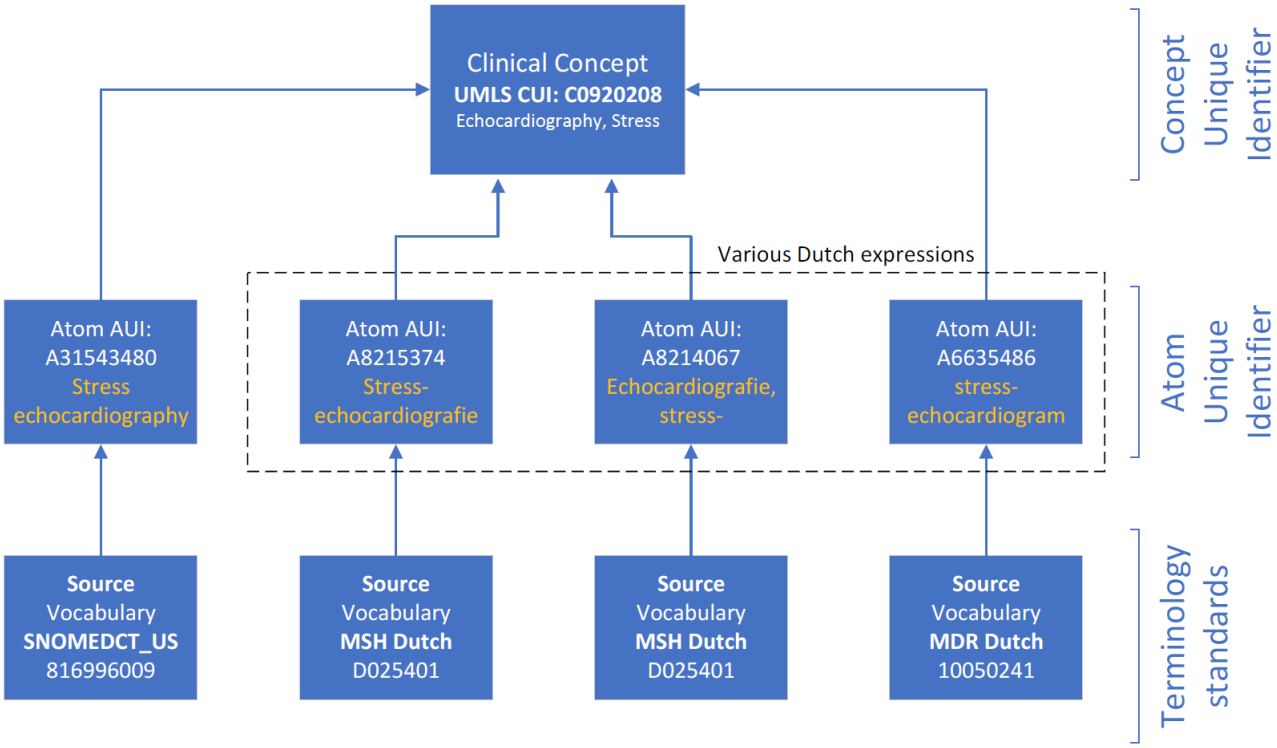

Figure 6: The UMLS Concept

terminology server solution of Metke-Jimenez et al. [2018] can offer a possible solution to translate locally defined clinical concepts into internationally recognized terminologies. Moreover, the terminology solution can facilitate the management to keep locally deployed terminology solutions up-to-date. Diomaiuta et al. [2017] proposes a software architecture that makes practical use of such a terminology solution. The architecture assists healthcare professionals in automatically translating registered codes toward their corresponding textual descriptions. The proposed system architecture can read and store clinical information efficiently from and to clinical documents stored on a health server to fulfill semantic interoperability. The study aims to represent a foundation for building more complex interoperability solutions using the FHIR standard. The research conducted by Diomaiuta et al. [2017] is limited to a software architecture capable of unambiguously representing a minimal clinical dataset. Furthermore, the solution does not integrate with other existing architectures, eliminating complex integrations with realistic medical environments.

Various programs are ongoing in the Netherlands to promote the unambiguous registration of clinical information[10]. Because unambiguous registration requires clear agreements about the precise registration of clinical terms, hospitals move away from free text registration to apply predefined clinical terms. An example is the Dutch multidisciplinary administrative body (Dutch Hospital Data[11]) committed to setting up standardized vocabulary listings containing relevant clinical concepts to assist clinical practitioners dur-

[10]https://www.nictiz.nl/
[11]https://www.dhd.nl

ing registration activities. We recognize the development of a single unified language to promote unambiguous registration. This uniform language expressed as 'Diagnosis thesaurus[12]' and the 'Verrichtingen thesaurus[13]' ensures that medical practitioners can unambiguously register all patient treatment activities [Kieft et al., 2017]. These registration standards embrace various clinical specialties spanning diverse mappings with international terminology systems (including cardiology) to promote international data exchange. Since Dutch EHR suppliers agree to implement these registration standards, there are agreements about registering health information in the Netherlands to facilitate the exchange and promote reuse of healthcare data, called 'registration at the source'[14] [Simons, 2019]. Additionally, it reduces the registration burden for clinical practitioners and accelerates scientific research by unambiguously registering clinical concepts at a national level. All these programs aim to promote a uniform clinical data representation independent from the EHR vendor.

Because leading healthcare organizations address innovation by delivering scalable and interoperable health information management solutions connecting data between various in-house developed software solutions seamlessly, medical software development focus on continuous evolution [Pascot et al., 2011]. To promote this, leading healthcare organizations developed digital health ecosystems to decouple healthcare data from cross-business software solutions to reduce integration efforts [Gopal et al., 2019]. Software developers can subsequently build new software solutions on top of the ecosystem, serving as a collaborative platform for accelerating cross-business integrated care solutions' software development. External product integration with the healthcare ecosystem using third-party software adapters maximizes health data benefits enabling collaboration. Scaling up the healthcare ecosystem by bringing multi-vendor health data together accelerates innovation by making healthcare data enterprise-widely available in a secure ecosystem of information and services according to a pre-defined common data model [Weir, 2019]. Figure 7 depicts a schematic overview representing a standardized integration of various software products towards a common data model. Since FHIR provides the flexibility to model clinical information in many ways, we need to shape core FHIR resources to optimize their use in a particular setting. This modeling mechanism is called FHIR profiling. Various studies describe applying the FHIR profiling mechanism to achieve semantic interoperability with other software systems [Semenov et al., 2018] [Gulden et al., 2021]. We also recognize the FHIR profiling mechanism's appliance to describe clinical information models promoting clinical data's reusability in healthcare [Kieft et al., 2017]. We discuss this further in subsection 2.3.

2.2. Health data extraction

Because CDRs store data in various formats according to specific terminology vocabularies, the collected information needs to be transformed towards a standardized target repository to represent a uniform data structure conforming to a predefined data model. Ong et al. [2017] discussed an approach for health data extraction to provide a scalable solution

[12] https://www.dhd.nl/producten-diensten/diagnosethesaurus/paginas/diagnosethesaurus.aspx
[13] https://www.dhd.nl/producten-diensten/verrichtingenthesaurus/Paginas/Verrichtingenthesaurus.aspx
[14] https://www.registratieaandebron.nl/

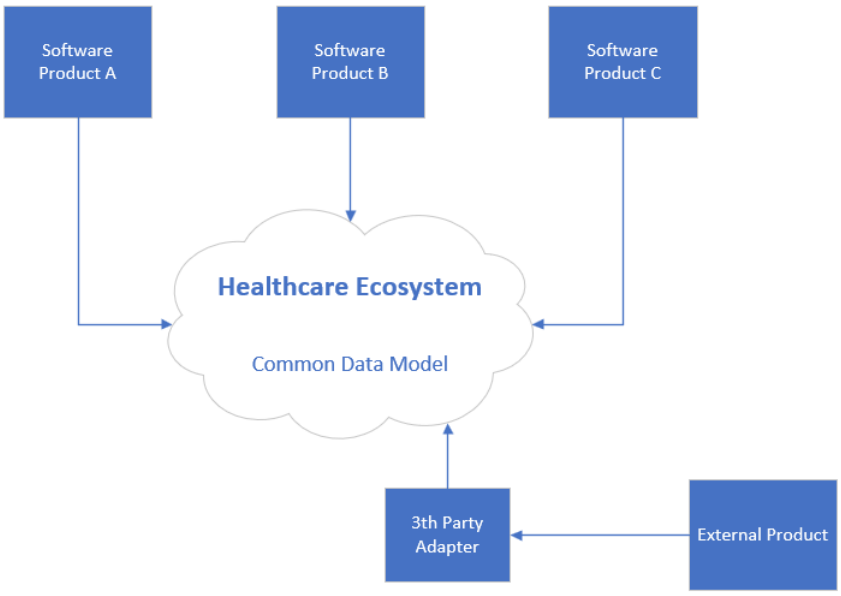

Figure 7: Standardized data model

harmonizing clinical data using the extract, transformation and load (ETL) process to lower the technical burden and simplify the transformation process. ETL is a procedure to copy data from one or more CDRs towards a target repository which represents the data into another context compared to the source location. Ong et al. [2017] developed a dynamic ETL (D-ETL) solution to extract data from multiple data sources and transform it into a common format according to a pre-agreed standard terminology that meets business needs. The provided approach offers a flexible solution addressing critical technical challenges of the ETL process such as compatibility, scalability, data quality and error handling.
Given the diversity in data representations caused by the originating CDRs, a *compatibility* challenge exists to transform the data into a commonly agreed target data model. This transformation is achieved by an integration engine, generating SQL statements based on existing transformation rules. The study reports that a great deal of knowledge and clinical experience of the data structures is required to understand correctly which source data elements correspond to the target data elements. Data integration tooling can help to realize data mappings but is limited in functionality to address complex transformations.
Another challenge that Ong et al. [2017] faces is *scalability*, since clinical research systems must be able to cope with the volume of requested health data and variety in CDRs across different clinical research networks. The application of D-ETL rules, defining how to map and combine source data fields to the target data format, shows that mappings can be efficiently shared and maintained across several teams sharing the same source and target data formats. Efficient management of data transformations configurations is a valuable feature that we should investigate further in this study to optimize the implementation effort between various healthcare settings for known data transformations.
Finally, *data consistency and error handling* plays a challenging factor in the ETL process. Source data might be inconsistent, and conflicting data sets need to be identified as they can affect the efficiency of the ETL process. The study showed that data validation is crucial immediately after the data extraction process from source repositories to ensure the

success of subsequent steps. Ong et al. [2017] concludes that this process is very time-consuming. SQL queries can become complicated and require clinical, technical and specific knowledge of the CDR data structure to validate conflicting data elements before the source data elements can be stored in the target database.

Besides healthcare, the previously discussed ETL technology is also applied in other sectors, such as logistics, as business conditions in the global market require fast and efficient decision-making solutions [Radivojević et al., 2013].

2.3. Reusability of Healthcare Data

The healthcare environment is a complex and dynamic system containing multi-vendor clinical data repositories. Since digitization concerns various healthcare aspects, healthcare organizations need to flexibly deal with continuous changes in healthcare providers' underlying IT infrastructure to deploy software solutions flexibly. Model-driven engineering methodologies can introduce standardization to tackle this problem since standardized information models aim to increase productivity during the software development process by reusing standardized data models [Schmidt, 2006]. However, implementing a common data information model around different repositories remains a challenge in healthcare [Demski et al., 2016].

Investigation of Marco-Ruiz et al. [2015] is related to the reusability of clinical data stored in EHR systems. Richen and Steinhorst [2005] express it as harmonization, which prevents or eliminates differences in the technical content of standards with the same scope. Marco-Ruiz et al. [2015] focuses mainly on challenges in semantic interoperability and the interoperability of different existing data models. The aim is to represent clinical data as a uniform standard across multiple healthcare institutions, maintaining clarity about the applied terminologies to avoid interpretation issues.
The discussed openEHR[15] standard is currently a promising approach to address those challenges built upon the two-level modeling approach using a reference information model to represent clinical data. The two-level modeling approach allows the development of a technical infrastructure entirely independent of the underlying clinical data model, described as archetypes. Those archetypes form the main building blocks to model the clinical data structures and describe a template defining the data representation and terminology binding. The representation of clinical data, according to those templates, improves semantic operability across different repositories. Marco-Ruiz et al. [2015] developed a data warehouse architecture, based on openEHR archetypes, to address interoperability issues. They provided a vendor and technology-neutral approach to query healthcare information systems spread over multiple healthcare institutions. The proposed solution offers remarkable advantages as applications can be developed around a centralized artifacts repository containing openEHR compliant data. Several pilot studies show some investigation using the openEHR standard and report challenges to agree on an internationally accepted clinical data model [Haarbrandt et al., 2016] [Min et al., 2018]. Further investigation reports limited research on a national and large-scale level [Li et al., 2018]. Some national development initiatives are ongoingly related to the adoption of the openEHR standard. Still, due

[15] https://openehr.org/

to the variety of the market and the lack of well-defined clinical information models, there are no national agreements yet in Belgium and the Netherlands [Pedersen et al., 2017].

2.4. Interfacing engine

Accomplishing interoperability is crucial to fulfilling the goal of a fully integrated landscape, delivering a common data structure against multi-vendor CDRs. Today, different integrator engines are available to achieve interoperability between different medical repositories. As this research collaborates with Philips Research and Philips partnered with Orion Health to team up interoperability for all business units, we aim to implement the solution on the same interoperability platform. The Orion Health Rhapsody Integration Engine[16] enables a high-performance, healthcare standard-based integration engine recognized as a global health informatics solution [Binobaid et al., 2016].

2.5. Related Work Conclusions

In this subsection, we want to argue the conclusions taken based on the earlier reviewed literature. This literature review should outline the foundation for our further research.
Having access to clinical information at all times is essential to provide optimal care for patients. Since we aim to develop a solution that can unambiguously represent cardiovascular data at all times, the previously discussed HL7 v2 and v3 standards expose shortcomings. The unstructured HL7 v2 standard and the high degree of customization make this standard inappropriate in a multi-institutional environment. Moreover, the flat-file structure makes it challenging to interpret messages. The successor HL7 standard (HL7 v3) introduced a reference information model to offer more standardization and making information easier to interpret but lacks recognition in Belgium and the Netherlands because of its complexity. On the other hand, the Internet-based FHIR standard offers a common ground to move away from proprietary data representations to exchange clinical information. Detailed FHIR implementation guidelines published on the Internet empower software developers to develop software applications across various platforms. Additionally, the HL7 FHIR interoperability standard provides atomic data access through REST web services, eliminating the need to share entire messages in contrast to older HL7 v2 and v3 standards. By giving software developers access to clinical data through an FHIR API, we foresee significant benefits related to performance and accessibility. Although we believe FHIR is a step in the right direction to enable unambiguous data exchange between different software systems, the solution to true semantic data interoperability does not seem to be for tomorrow. Due to the enormous variation present in the collected data, there is no consensus on a clinical concepts' data model. In our opinion, FHIR can offer an excellent foundation to support semantic data interoperability to ensure a similar clinical data interpretation at any location. Additionally, the REST API architecture's simplicity makes data easily accessible through modern web technology, enabling software developers to quickly and easily access clinical information from any device. Still, by only using the FHIR standard, we cannot solve the entire interoperability issue.
For this reason, we started looking for solutions that can contribute to solving the semantic

[16] https://orionhealth.com/global/strategic-partners/rhapsody/

problem using the FHIR standard. The solution provided by Metke-Jimenez et al. [2018] describes an exciting and useful solution to address this problem. The study presents a terminology solution based on the FHIR standard, making it possible to access different internationally recognized clinical terminologies through an API call. The solution also considered providing a centralized mechanism to keep the terminology server updated with the latest release of clinical vocabularies. Within the context of our research, it seems reasonable to include this feature to minimize implementation effort and maintainability of clinical terminologies. A possible obstacle to this terminology solution is that the terminology system cannot deal with locally applied terminology standards. Further research should determine how we can deal with this efficiently.
Diomaiuta et al. [2017] proposes a software architecture that makes practical use of such a terminology solution. This study forms an excellent foundation to get a clear picture of the software architecture, taking into account semantic interoperability. However, this research lacks any form of integration with other software platforms. As a result, we have to investigate further how we can integrate an extensive dataset from various data repositories within the proposed software solution. It is also essential to note we need to explore a solution deployable in a multi-institutional setting.
Ong et al. [2017] answers this integration issue very concretely and offers a solution entirely aligned with our expectations. We aim to elaborate further in our study on the applied dynamic-ETL methodology because it is valuable for collecting and transforming efficiently various data elements towards a predetermined target data model.
Consequently, we explored several literature studies on how to model the extracted clinical information unambiguously. Ultimately, we want to model the clinical concepts in an unambiguous way to promote the clinical concepts' reusability in different medical environments. Ideally, we are looking for a universal set of clinical models. The literature search results show that reality is unfortunately different since we can model clinical data in a dizzying array of compositions that include their characteristics and rules. We investigated the differences between FHIR and the openEHR standard. After comparing both standards focusing on data interoperability, we can conclude that the FHIR standard is better suited for customization in a standardized way without profiling a fixed clinical data model. Additionally, FHIR offers a flexible solution for software developers to exchange medical data through user-friendly and straightforward REST APIs. OpenEHR, on the other hand, is more focused on the persistence of data, without space for customization in the data model. Since we aim to obtain an unambiguously and enterprise-widely defined clinical dataset accessible for cross-business software developers, we assume the FHIR standard provides a better foundation to build innovative patient-centric applications supporting clinical data exchange.

3. Problem Analysis

Cardiac catheterization is a medical procedure to diagnose and treat certain cardiovascular conditions. During the medical examination, a long thin tube, called a catheter, is inserted into an artery of the groin, neck, or arm. The catheter is guided through the blood vessels of the patient to the patient's heart. Next, the catheter is used by the performing physician to perform diagnostic tests and treatment. The data flow of a heart catheterization procedure consists of several processes addressing each a particular phase of the complete examina-

tion. A complete cath documentation procedure exists of five parts, defined as mission briefing, procedure logging, creation of the physician report, providing the inventory for billing and registry, and patient debriefing as depicted in figure 8. The first documentation process addresses the mission brief to prefill and provide a single clear overview of the essential patient-related information. Essential information contains patients' medication history, lab values, medical data about previous studies, and administrative information registered into the EHR of the hospital. The mission-brief process should give the physician a complete picture of the patient, which is crucial for the physician to prepare for the cardiovascular procedure.

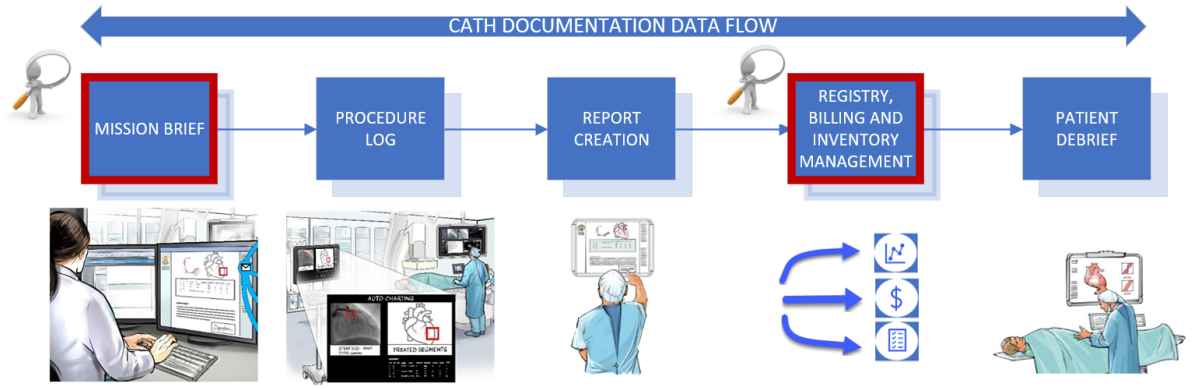

Figure 8: The complete cath documentation process

Once the physician collected the crucial information, the procedure can start. During procedural logging, nurses track essential events and material usage of the examination (e.g., location of stents, functional measurements, and used materials). After finishing the registration of all vital parameters and materials into the cardiovascular information system, the physician can create and finalize the patients' report containing a summary of the catheterization procedure. The next process is a more time-consuming process, related to billing and national registry reporting, to guarantee and monitor the quality of the cardiovascular procedures. Billing uses mainly the physician report and material usage as input, while registry reporting requires a mandatory dataset imposed by national registry organizations to improve the quality of care, such as the National Cardiovascular Data Registry[17] (NCDR). The last phase of the cardiovascular procedure is the patient debriefing phase. The physician informs the patient about what is happened during the examination and hands over a printout to take home. These documentation processes have to be done manually and require considerable effort from the medical staff.

This study focuses on the mission brief and registry management process within the complete cath documentation process, depicted red in figure 8. The intention is to develop a scalable solution to collect automatically diverse multi-source patient-related cardiovascular data consistently in a pre-defined target data structure allowing efficient and straightforward data manipulation. We define scalability as the possibility of deploying the solution in various multi-vendor hospital environments in Flanders or the Netherlands. In

[17] https://www.ncdr.com

addition to multi-institutional deployability, we define scalability as the ability to make the solution future-proof in a relatively straightforward manner to process new clinical concepts efficiently with minimal implementation cost. Today's use cases consist of creating a centralized dashboard for cardiologists and delivering a pre-defined dataset to an external third-party registry entity for quality validation. In practice, healthcare companies envision this problem on a much broader scale to facilitate the development of medical software applications. The red highlighted process blocks in figure 8 refer to the positioning of the use cases into the cath documentation process. Currently, physicians need to review the previous relevant imaging and diagnostic studies to get a comprehensive view of the patients' history. Our solution results in a less labor-intensive documentation process letting cardiologists better prepare for the clinical procedure without the administrative burden to collect the required information manually. Technically, the solution must contribute to shortening the needed integration time across diverse medical institutions improving cost-efficiency. Additionally, scalability fulfills an essential role in delivering a future-proof solution capable of collecting broader data sets.

Figure 9 represents the diversity depicting the differences between medical institutions. Hospitals can contain various clinical data repositories from different vendors holding clinical data represented according to locally applied coding terminologies (represented by the different colored clinical source systems). Furthermore, clinical repositories can offer their data in different ways, which can differ per vendor and medical institution, and where external factors can be decisive. Some important CDRs, depicted in figure 9, are the laboratory information system (LIS) helping medical laboratories managing the laboratory results and related documentation. Further, we depicted a PACS (Picture Archive and Communication System) system representing a software system to store and diagnose clinical imaging files. A PACS system's primary purpose is to manage all captured medical images efficiently according to the patients' history. In addition to the EHR, we also illustrate a Cardiovascular Information System (CVIS) acting as a data management platform optimized to the cardiovascular physician's needs.

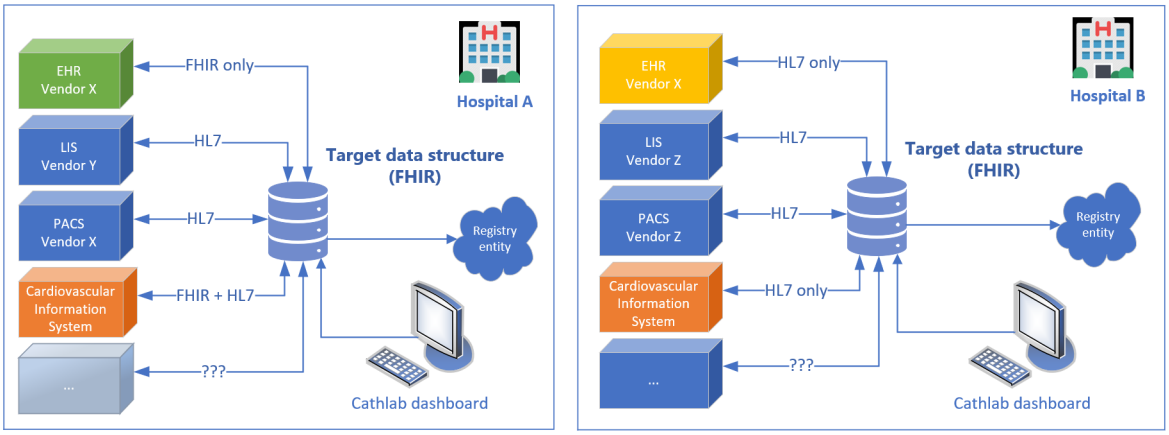

Figure 9: Variety in clinical data sources across different medical institutions

4. Research Methodology

4.1. Research Objective

Medical organizations continuously produce large amounts of clinical data during daily activities captured in many different facilities and multi-vendor systems. To empower healthcare organizations to continue improving patient care by developing innovative software solutions, they need to gain powerful insights from these independent clinical data repositories. Unfortunately, the lack of standardization across these various CDRs causes these insights to be lost. For this reason, leading healthcare organizations are looking for an efficient integration solution to collect and transform multi-sourced clinical data into unambiguously defined clinical data concepts. This solution empowers large healthcare organizations to interpret clinical information in an unambiguous way delivering value to various innovative software systems. However, healthcare organizations observe that realizing such an interoperability solution is challenging because of the diversity across medical institutions. Because medical institutions control their own data workflow in a multi-vendor setting, each integration project demands a lot of customization, resulting in high integration costs making healthcare IT projects unprofitable.
This study aims to tackle the existing variety in clinical data repositories and applied data representations across different healthcare institutions, dealing with challenges to collect data efficiently into a standardized target data format. Section 2 earlier outlined in more detail the extraction process and highlighted critical technical challenges to be considered. The goal is to provide an efficient solution that significantly reduces the data integrators' configuration effort needed to collect and translate multi-sourced cardiovascular data concepts towards an interpretable standardized data set. Additionally, we aim to estimate the amount of time required to implement the prototype in another clinical setting. Furthermore, we want to estimate the amount of time needed to expand the designed prototype with a definite number of clinical concepts. These time indications should enable healthcare organizations better to estimate the efficiency of integration projects in advance and contribute to further optimize the scalability of the integration solution by revealing inefficient extraction and data transformation processes.

4.2. Research Questions

We aim to design a software architecture to efficiently prepare a standardized and interpretable dataset containing accurate and consistent information from various subsystems. To realize the collection of multi-sourced clinical data, we need to have a clear overview of *what* we exactly want to represent into the target datastore. Based on a specified dataset, which can be subject to variation, more investigation is required *where* exactly to find the corresponding data in various CDRs inside the healthcare organization. Knowledge about the location and the data representation makes it more convenient to understand *how* to approach the different data sources.

An essential part of this research is finding an efficient solution that is deployable across different healthcare institutions. To accomplish this aim, we would like to know if all necessary clinical data elements are available in the source repositories and can be interpreted consistently through our solution. It is essential to know how many of the required data is

represented in a structured format to ensure the unambiguous representation of the source data, making further processing possible. In case the source dataset is ambiguously defined (e.g., unstructured), the transformation to the target data model will become much more complicated and requires other resolving technologies. Because there is no consensus yet defining an internally standardized target data model within Philips, we refer to a fictitious data model to shape the information correctly according to the FHIR specification standard. Although we consider the retrieval of unstructured data out of this research's scope, it is useful to know which fraction of data we find in a structured and unambiguously defined format in diverse CDRs. This information can help us further evaluate the solution's efficiency if we only focus on structured data.

As discussed in section 2, *standardization* is an important concept to transform clinical data consistently across diverse clinical environments. To achieve transformations deployable at different healthcare institutions, we want to find similarities in local data representations [Dykes et al., 2010]. To reduce the amount of work required to transform the data while retaining the clinical context, we want to know if we can find some *patterns* related to the data representation of specific clinical source systems. Finding those patterns can boost the efficiency of the transformation process using predefined transformation sets. The obtained knowledge should enable us to get a clear overview of reusable data transformations applicable within the software solution. Subsequently, we want to investigate how to define and manage data transformations with minimal effort to improve our solution's scalability.

To match real situations, we want to set up a prototype to test our solution on a medical institutions' test environment. Furthermore, this study aims to deliver a prototype built on the existing integration platform deployed worldwide across Philips' integration portfolio. The integration platform is mainly responsible for consolidating the extracted information from the appropriate source repositories so that software developers can have rapid access to unambiguously represented cardiovascular information. Since Philips already developed standards-based interoperability solutions between patient care devices and hospital information systems, we want to investigate the extent to which we can reuse existing software components of these solutions within our prototype. Software reusability defines reusing existing software components within the software development process to develop new software. The involvement of existing software components empowers us to minimize quality issues and drastically reduce the time required for the prototype development process [Mateen et al., 2017].

To conclude this study, we would like to evaluate the developed solution in a clinical test environment. Implementing this solution will give us some understanding of how the prototyped solution will behave in a real clinical environment. After this final implementation, we want to report a conclusion about the number of data elements collected successfully from various source data repositories. Philips employed clinical scientists with the necessary clinical background and experience are engaged to assess the correctness of the data transformations. To validate the solution's efficiency, we refer to a subset of software quality characteristics explained in the ISO-9126 standard [18] and literature focusing on how to

[18] https://www.iso.org/standard/22749.html

measure software product quality [Jung et al., 2004] [Theodorou et al., 2017]. As a result, we concentrated on some key indicators to determine the efficiency of the software solutions, including:

- Efficiency - *time behavior*. Representing the time required to fulfill the entire ETL process.
- Efficiency - *scalability*. Representing the extent to which the solution is future-proved to extract more clinical concepts in a multi-institutional setting.
- Maintainability - *changeability*. Representing the effort required to add additional clinical concepts.

Based on this research objective, we formulate the overall research question.
[RQ] ***How to design a scalable and multi-institutional deployable data integration solution for cardiovascular data?***

To answer this question, we split the overall research question into several sub-questions.

- [SRQ1] Which clinical data repositories could be queried for cardiovascular data collection?
- [SRQ2] How to design an extraction method for cardiovascular data collection?
- [SRQ3] Which variations in data representation can be found across the different medical institutions?
  - [SRQ3a] How is cardiovascular data represented in various clinical environments?
  - [SRQ3b] Which fraction of the cardiovascular data is in a structured format?
- [SRQ4] How to realize data transformation efficiently for expressing the cardiovascular information in an interpretable fashion?
- [SRQ5] How efficient is the proposed software prototype?
  - [SRQ5a] Which software metrics could be used to express the efficiency of the implemented software prototype?
  - [SRQ5b] What is the impact of the proposed software solution?

4.3. Research Approach

To achieve the aim of this research, we will apply the Design Science Research Methodology for information systems research by Peffers et al. [2007]. It is a commonly accepted framework to model the life-cycle of design science research. The process is composed of several steps which define the design of the entire research methodology. Based on this framework, we will classify the applied research methodology for this study into different sub-processes, where we will explain the various activities referring to the corresponding research questions.

Problem identification and motivation. Collecting patient-related information from multi-source clinical data repositories is a labor-intensive task for a cardiologist to prepare

optimal for a cardiovascular procedure. To support cardiologists collecting and organizing data for evidence-based decision support, we want to investigate how to leverage, with minimal effort, the variety of healthcare data among different healthcare institutions (see [SRQ1], [SRQ2], [SRQ3]).
The research is based on the collection of a predefined dataset representing crucial information for cardiovascular procedures. A reference target dataset is delivered by Philips but is subject to variation and is required to answer [SRQ1]. To answer this research question of which clinical repositories we need to query, we have to evaluate which source repositories exist into several healthcare institutions' IT environment holding a subset of the initial target dataset. A clear overview of the various existing clinical data repositories is required to understand the medical IT landscape's variety, contributing to how we can retrieve this data [SRQ2]. For this research, we are exclusively interested in the data representation of structured datasets and the applied vocabulary used to codify clinical data [SRQ3a]. Based on earlier information about the representation of data, we are interested in finding some patterns in applying specific terminology systems for particular data sources in hospitals. Pattern interception can improve the efficiency of our solution by implementing predefined logic for recurring patterns. Possibly, we can find some exciting differences applied on a national level or some pattern matching based on the vendor of the software application. After identifying structured cardiovascular concepts in various CDRs, we want to determine which fraction this represents from the intended Philips data set [SRQ3b]. This information can give us a better understanding of the amount of cardiovascular data available in a structured manner across various CDRs.

Definition of the objectives for a solution. The objective of this study is to build a *software architecture* and *prototype* to reduce the configuration effort as a result of the variety in healthcare data among medical institutions (see [SRQ4]). Significant challenges are the diversity in data representation between various clinical data repositories and the requirement to represent all collected data in a standardized target data structure for unambiguous interpretation. The target data structure can serve as a supporting entity providing effortless access for software development activities or reporting efficiently patient-centric quality measurements towards external quality organizations, ensuring evidence-based cardiovascular care.

Design and development. This research's delivery consists of a software architecture proposal and a prototype for evaluating the proposed solution [SRQ4]. To propose a final solution, the design of the software architecture should include four main components.
The first component is responsible for extracting the clinical data out of the different CDRs. To query each data source repository, we need to find out how to access the data. The earlier investigation, described in the related work section, can be used to answer [SRQ2]. Additional research is required to investigate applicable extraction methodologies available in hospitals. Knowing the most commonly used methods forms an excellent foundation to continue designing the software architecture proposal. The second component of the software architecture handles the variations related to data representations into all CDRs and focuses more on maintaining semantic data interoperability [SRQ3]. The software architecture design should send predefined queries to specific clinical repositories to collect all the required information. The solution should map and process the data according to

terminology rules, specified by a terminology server, being part of the architecture. The terminology server should facilitate the required logic to transform all data elements towards a predefined target data structure as discussed by Metke-Jimenez et al. [2018] in section 2. All transformation tasks should be realized using the Rhapsody Orion Health[19] interoperability platform, worldwide deployed by Philips managing their interoperability solutions. The interoperability platform's presence makes the solution relatively easy to deploy with existing Philips customers to streamline with other existing interoperability solutions. We also need to keep efficiency in mind during the architecture design by managing eventually recognized applicable patterns to realize particular transformations. After the data extraction and transformation, the third component of the architecture should store the data elements into a predefined target data structure. Specific extracted fields may need to be transformed and combined to conform to the target data model. We must investigate how to realize these transformations in practice as efficiently as possible to increase scalability. To lower the development effort, we will analyze which existing software components of the integration engine are valuable to collect specific data from the CDRs. For testing purposes, we will use a docker[20] image to simulate our personal FHIR test implementation server for development and testing. This image is built on HAPI[21], which is an open-source implementation of the FHIR specification written in Java. It is important to note that this test environment is only suitable for prototyping without ensuring scalability and performance. To ensure patient-related information's privacy and confidentiality in the test environment, we will use Synthea[22] as an open-source patient generator to simulate realistic Protected Health Information (PHI) data. When developing the prototype, we also have to consider some essential non-functional requirements (NFR). Important NFRs, defined in the ISO-25010[23] model, are interoperability, functional correctness, functional completeness, scalability, and reusability. Interoperability addresses the current compatibility issues introduced by the variety of data representations existing in the different CDRs. We must accomplish the multi-source data extraction preserving the original data's meaning to represent the required complete data set. To facilitate the logic needed for a correct data transformation, we need access to a terminology solution, as described earlier in this section. Functional correctness and completeness require some validation and should be handled by the fourth principal component of the solution, the validation part. The remaining NFRs are more related to the validation process and discussed in the Design Science Research Methodology's validation activity.

A preliminary scheme of the software architecture, representing the above design and development process, is depicted in figure 10. The red and green entities correspond to the principal software components of the architecture. The green entities represent eventual existing components within the Philips integration engine, while the red ones represent components that are yet to be developed. The Rhapsody configuration module corresponds with the integrated development environment to build and configure all interfacing and data processing components. Finally, the Rhapsody monitoring component can track all the processed messages via a web interface.

[19] https://orionhealth.com/
[20] https://www.docker.com/
[21] https://hapifhir.io/
[22] https://synthetichealth.github.io/synthea/
[23] https://www.iso.org/standard/35733.html

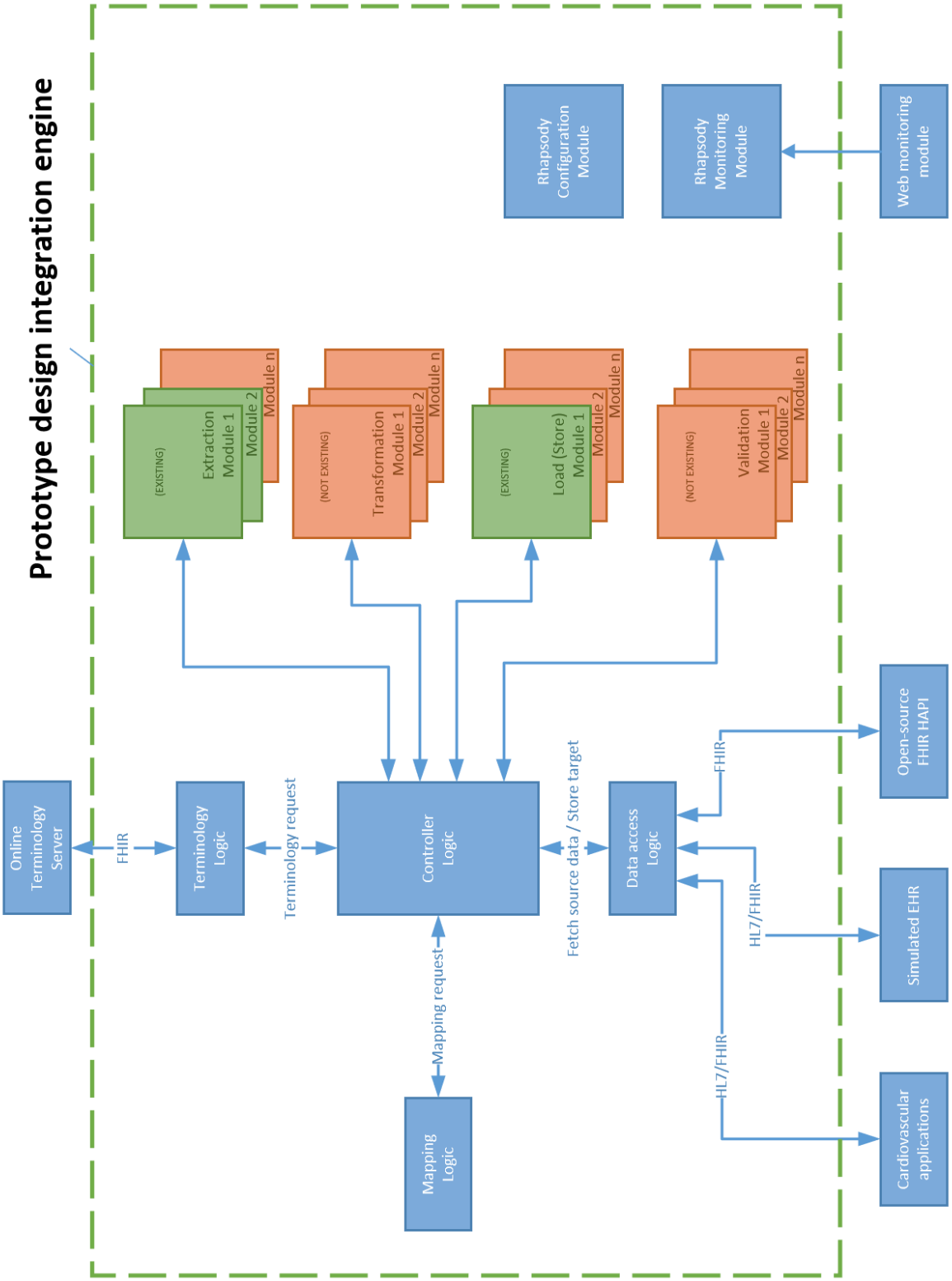

Figure 10: Prototype design

Demonstration. Developing a prototype can give us more beneficial insights to better understand any shortcomings or inefficiencies. First, we will set up a test environment in the Philips lab to develop and test the prototype. Into this test environment, we should simulate an EHR and some additional clinical repositories containing cardiovascular resources to reflect reality. The simulation in the test lab will only cover existing Philips products. Meanwhile, implementing the prototype in a customer environment can add additional value to simulate a more diverse environment. Consequently, we should implement the prototype in a clinical test environment inside one medical institution located in Belgium (Flanders) and the Netherlands. The option to apply the prototype in both countries has to do with potential variations we want to discover and evaluate at the national level. In the context of this study, it was not straightforward to get the support of two clinical institutions willing to cooperate. The current Covid-19 pandemic plays an essential role in this. As a result, we collaborate with an innovative Belgian healthcare institution to design the prototype, and subsequently, we can rely on another Belgian reference hospital to assist during validation testing.

Validation. The collection of clinical data from various source repositories is based on the principle of the ETL process. An inefficient transformation process can lead to the representation of incorrect values for cardiologists and affect the solution's scalability. To evaluate the solution, we will split the validation process into three parts. First, we want to compare the obtained results with the values of the original data elements to validate the correctness of the extracted data. Next, we will determine the evaluation criteria based on some metrics that can be applied to assess the efficiency of the solution [SRQ5a]. The intention is to measure the amount of time required to collect cardiovascular data in a medical test environment from multi-source clinical data repositories and represent it into an unambiguously defined clinical dataset. Additionally, we will provide a mechanism to estimate the amount of time it will take to implement the prototype in a different clinical setting and estimate the time it will take to scale up the prototype to process a broader range of clinical concepts. To conclude the validation part, we should evaluate all obtained results to estimate the impact of the proposed software solution [SRQ5b].

5. Solution Preparation

One of this study's main goals is to analyze which fraction of the cardiovascular clinical concepts defined in the earlier mentioned NCDR dataset we can collect from various clinical environments. The ultimate goal is to achieve this by finding a multi-institutional deployable solution to reduce the implementation time required to collect cardiovascular concepts. Having a mechanism to collect and represent extracted cardiovascular data in a standardized and unambiguous way could significantly benefit software development activities relying on unambiguously defined cardiovascular information because it separates the technical design from the clinical concerns [Christensen and Ellingsen, 2016]. It empowers software engineers to develop software applications independently from the underlying (complex) clinical data model, usually unknown territory for software developers.

To better understand the relevant clinical concepts regarding Percutaneous Coronary Intervention (PCI) procedures, treating the narrowing of coronary arteries, we use an American data set (cathPCI dataset) used for government registration purposes. We refer specif-

ically to this American dataset because we want to conduct similar research beyond the Benelux on a global scale. Since the American cathPCI reference dataset provides a clear overview of PCI studies' relevant parameters, we consider possible variations in PCI practice patterns between the Benelux and the United States [Inohara et al., 2020]. Despite this diversity, the dataset provides an excellent reference to potential relevant parameters in which doctors are interested in determining a patient medical condition. A detailed description of these clinical concepts defined in the cathPCI dataset is publicly available on the Internet [24].

A logical next step is to find out where exactly we can find the required information. In this section, we want to answer the sub-research question one [SRQ1] to clarify which clinical information sources we need to query to collect clinical concepts relevant for cardiovascular examinations. As discussed earlier in section 3, we must consider being able to deal with a multi-vendor system environment storing each clinical information according to their own proprietary data format in a possibly unstructured way. Assuming we also need to consider our software solution's multi-institutional deployability, we soon realize that we have to look for another efficient solution to obtain cardiovascular data. Furthermore, we see that due to the enormous increase in digitization of medical data and the requirement to report data to government registries for quality control, hospitals accommodate centralized information systems platforms to collect heterogeneous data. Healthcare professionals consult these information systems to overview electronic health records captured from heterogeneous systems [Steinhubl and Topol, 2015]. These information systems improve care quality by enabling the automatic integration of clinical data recorded in various siloed data repositories. An example of such a centralized information system is an EHR system that contains an enormous amount of structured and unstructured data obtained from various healthcare systems. The availability of these massive amounts of clinical data offers enormous benefits for clinicians and can support scientific research. Those amounts of data can be used for various objectives improving healthcare, varying from developing medical software applications, clinical research, quality registration, clinical decision support Njie et al. [2015], etc. [Johnson et al., 2018].

As a result of the ZOL Genk data-driven mindset, we recognize a high degree of cardiovascular data integration into the EHR. However, we cannot assume that every medical organization integrates the same amount of clinical data within the locally deployed EHR software platform. The amount of health technology and software applications to register and analyze data within various hospitals introduces enormous technical and economic challenges for healthcare organizations implying that hospitals cannot always afford the same integration level with the EHR [Figueiredo, 2017]. As a result of the Covid-19 pandemic, we have not investigated the level of cardiovascular integration at other medical institutions. However, to capture eventually missing cardiovascular information integrated into the EHR, we aim to capture non-integrated cardiovascular information directly from diverse siloed CDRs capable of providing an HL7 v2 message stream holding cardiovascular data. We mainly envision cardiovascular information systems (CVIS), depicted in figure 9, as possible non-integrated CDRs holding crucial cardiovascular information. For this reason, we primarily focus on a software solution extracting cardiovascular data from EHR

[24] https://cvquality.acc.org/docs/default-source/ncdr/data-collection/cathpci_v5_codersdatadictionary_09172020.pdf

software platforms. Additionally, we still envision the ability to complete missing cardiovascular information in the EHR with clinical concepts captured through a second data extraction pipeline listening to HL7 v2 data communication.

5.1. Market Analysis

In recent years, we have seen rapid growth in the adoption of EHR software platforms in various medical settings, strengthening the centralized acquisition of vast amounts of clinical data [Nguyen et al., 2014]. Therefore, we can assume that the EHR acts as a single source of truth, integrating with individual siloed CDRs, making patient data hospital-wide accessible to healthcare professionals and patients.

Since the intention is to focus mainly on EHR systems to extract cardiovascular data in a multi-institutional environment, we first need to know which important EHR platforms are active in Flanders and the Netherlands. In this section, we want to outline this diversity by presenting our market analysis results. This market analysis is fundamental in determining the EHR vendors we should focus on to continue our research and answer SRQ1. To accomplish an overview of the market, we consulted Philips's installed base and entered into discussion with individual medical institutions and EHR suppliers. In figure 11 we represent a graphical overview of the EHR market share in Flanders, while figure 12 represents a similar overview for the Netherlands.

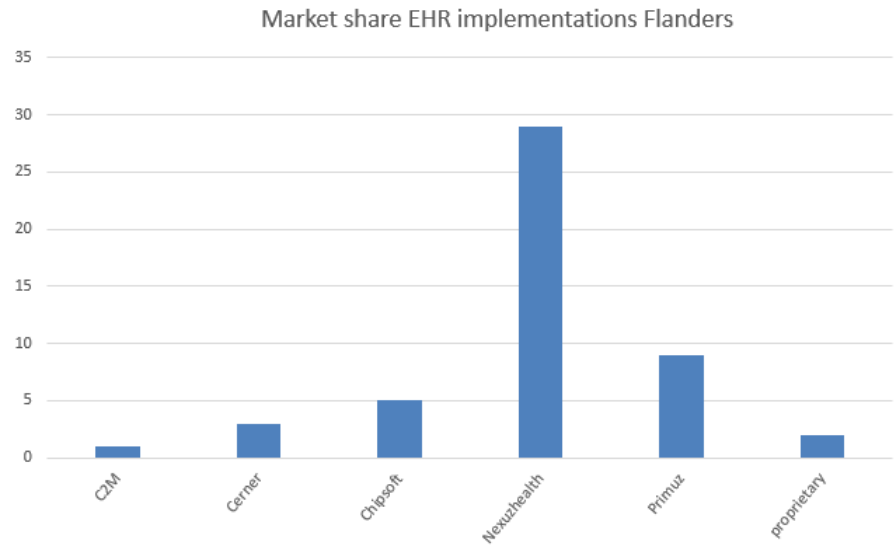

Figure 11: Market share EHR implementation Flanders

Figure 11 shows that Nexuzhealth is a substantial market leader for the Flanders region. Nexuzhealth is an EHR, originated from a medical partnership between the academic hospital Universitair Ziekenhuis Leuven and Cegeka, delivering the IT and consultancy services. Both partners combined their knowledge and developed an EHR platform offering software as a service (SaaS) to various healthcare institutions in Flanders. This SaaS platform aims to lower customer concerns and the EHR software platform's administrative burden by providing a SaaS delivery model to create an interoperability platform between dif-

ferent connected medical institutions, enabling efficient patient data exchange [Khan et al., 2012]. Sharing information across various medical institutions and unambiguously interpreting the clinical data is essential to fulfilling this software delivery model. The unambiguous identification of patient data encourages interoperability and forms a fundamental concept within this study. On the other hand, Universitair Ziekenhuis Brussel developed an integrated EHR software platform called Primuz. Besides Universitair Ziekenhuis Brussel, eight other Flemish hospitals use the Primuz EHR software platform to promote data interoperability between healthcare institutions. However, besides these two academic health institutions and other associated hospitals implementing their EHR software platforms, we observe that other large regional hospital groups in Belgium use ChipSoft[25] as their primary EHR supplier. Similar to healthcare facilities connected to one centralized SaaS platform, we recognize the need to exchange clinical information between collaborating hospitals using the ChipSoft EHR to optimize regional healthcare. Other hospitals in Belgium are still looking for a new EHR vendor and are negligible in this study because of their low market share. Since we are looking for a multi-institutional solution, we want to focus on EHR software platforms with the highest possible market share.

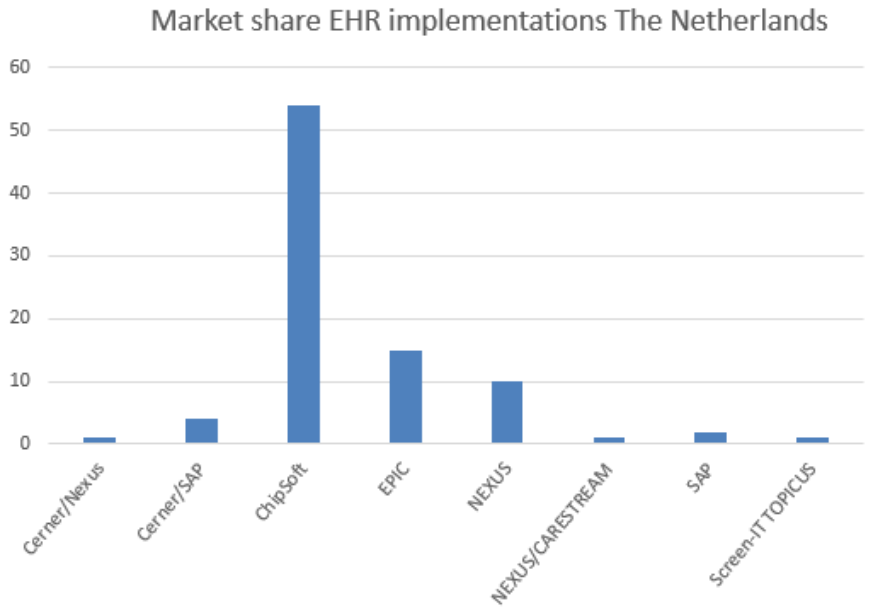

Figure 12: Market share EHR implementation The Netherlands

Focusing on the Dutch market, we see in figure 12 a completely different EHR landscape. In the Netherlands, the market is determined by two EHR suppliers. The Dutch market is dominated by ChipSoft and supplies 70% of the market. Besides ChipSoft, we see Epic[26] as the second leading supplier delivering EHR software solutions to hospitals. While ChipSoft is only focused on the Dutch and Belgian (Flanders) market, Epic is a global EHR software vendor.

[25]https://www.chipsoft.nl/
[26]https://www.epic.com/

We can conclude from the market research that there are several existing software vendors in both markets. Two major Belgian academic hospitals have actively contributed to developing their EHR platform and Nexuzhealth offers this software platform as a SaaS business model. Remarkably, only two EHR vendors are active in both countries. Other software suppliers only remain operational nationally. For the Netherlands, ChipSoft is the prominent EHR vendor, while Nexuzhealth is the leading EHR vendor in Belgium. ChipSoft is a leading Dutch EHR supplier, active in both countries delivering total healthcare software solutions for healthcare organizations. ChipSofts' software platform HiX[27] (Healthcare Information eXchange) is a hospital-wide information system managing electronic health records. Epic is the second-largest software supplier in the Netherlands. Remarkable, we see Epic infiltration in the Belgian market, but just outside Flanders in a French-speaking hospital. Interestingly, Epic is a global software provider that benefits from global innovation implemented within numerous medical institutions. The global presence makes further research attractive for Philips Research by innovating in a worldwide applicable solution. Nexuzhealths' EHR solution is only present in the Belgian (Flanders) market and holds 45% of the market share. Similarly, the Primuz EHR, developed by UZ Brussels, is only active within Flanders.

5.2. Approach Diverse EHR Software Suppliers

This research aims to develop a scalable solution that can be applied within Flanders and the Netherlands to extract cardiovascular data from different EHR platforms. We are currently providing an overview of the essential EPD suppliers present on the Flemish and Dutch markets through market analysis. Following the market analysis results, we have approached the most present EHR suppliers to obtain more information about the software platform's architecture and functionality. Contacting EHR vendors is challenging and refers researchers to medical institutions to discuss the architecture and functionality. EHR vendors are unwilling to share their architecture with third parties to maintain their dominant market share. Second, we must always consider privacy legislation regarding the use of medical data as EHR vendors do not own the medical data stored in their software platform. Unfortunately, we received rejecting signals from hospitals because no resources could be made available to elaborate on this study in detail due to the Covid-19 pandemic. In general, we noticed that EHR suppliers offering an EHR platform as a SaaS business solution were very closed to any form of collaboration. For one particular supplier, we were able to enter into a discussion. Ultimately, no follow-up has been giving for further participation in this research. The other SaaS provider did not respond at all to any form of request participating in this study. Disregarding the SaaS suppliers in Belgium, the most frequently present software suppliers are ChipSoft and Epic. ChipSoft, as a software supplier, gives the impression of being very closed. Due to a research agreement between Philips and the Belgian Ziekenhuis Oost-Limburg (ZOL) hospital in Genk, we agreed on an internship agreement to investigate how to approach cardiovascular data extraction from the HiX ChipSoft software platform in a scalable way. The close cooperation with ZOL Genk gives us the ultimate opportunity to test our proposed software architecture against our developed prototype. Consequently, we will refer during this research to the ChipSoft framework on which we will base our study. In contrast, Epic was willing to cooperate with our study but

[27] https://www.chipsoft.be/oplossingen/452/HiX

indicated to contact specific hospitals for further investigation. Due to the lack of resources within these hospitals, we could not continue this research for the Epic framework.

6. Solution Design

In this section, we would like to elaborate on the software solution design in response to the proposed solution's practical needs. Since we focus principally on EHR systems to extract cardiovascular information, we want to continue our investigation by answering sub-study question two [SRQ2] examining how to extract cardiovascular data in various clinical settings. To investigate how to extract cardiovascular information, we need to consider possible variations between different ChipSoft EHR implementations due to locally applied customizations made during the EHR implementation process in various medical organizations [Tutty et al., 2019]. However, since we aim to find a software solution to reduce implementation costs and make integration projects more profitable, we need to reduce the time required to extract cardiovascular data from various CDRs. Therefore, we are looking for a multi-institutional deployable mechanism to extract the data with as little effort as possible for healthcare integrators. Consequently, we want to design our software solution to represent the extracted data concepts from various ChipSoft EHR implementations in a standardized manner [Kruse et al., 2016]. In subsection 6.1, we will describe in more detail how we approach the data extraction mechanism. Secondly, we want to elaborate in subsection 6.2 on the first part of sub-research question three [SRQ3a], discussing the data representation variations of clinical information across different medical environments. Additionally, we outline how we will accommodate these variations in our software design. Finally, in subsection 6.3, we want to anticipate the design decisions on how to deal with the transformation of specific data concepts, which should be an excellent stepping stone towards our practical implementation. Figure 13 depicts a general conceptual overview of the proposed solution design to which we refer throughout the following sections.

6.1. Data Extraction

Healthcare institutions have large amounts of data available to support clinical processes, and EHR vendors offer various (customized) solutions to satisfy healthcare institutions' information needs. After a thorough analysis and several discussions with EHR vendors, we cannot find any standard method to extract cardiovascular data from various EHR platforms. Proprietary EHR tooling and specifically written SQL queries on data warehouses can offer possible solutions to meet the required (local) information needs. Data warehouse solutions are used by healthcare institutions and form a separate repository mainly used for reporting, research, and data analysis on EHR content [Karami et al., 2017]. A significant advantage of a data warehouse is that the production EHR environment's performance is not compromised since data queries do not take place on the production EHR database. Furthermore, data warehouse queries can be composed much more complicated than EHR tooling, as data warehouses can contain more detailed information [Karami et al., 2017]. Considering the data warehouse database synchronizes with the EHR production database at fixed times, there is no real-time alignment with the production environment. Additionally, developing a software solution to extract cardiovascular data using a data

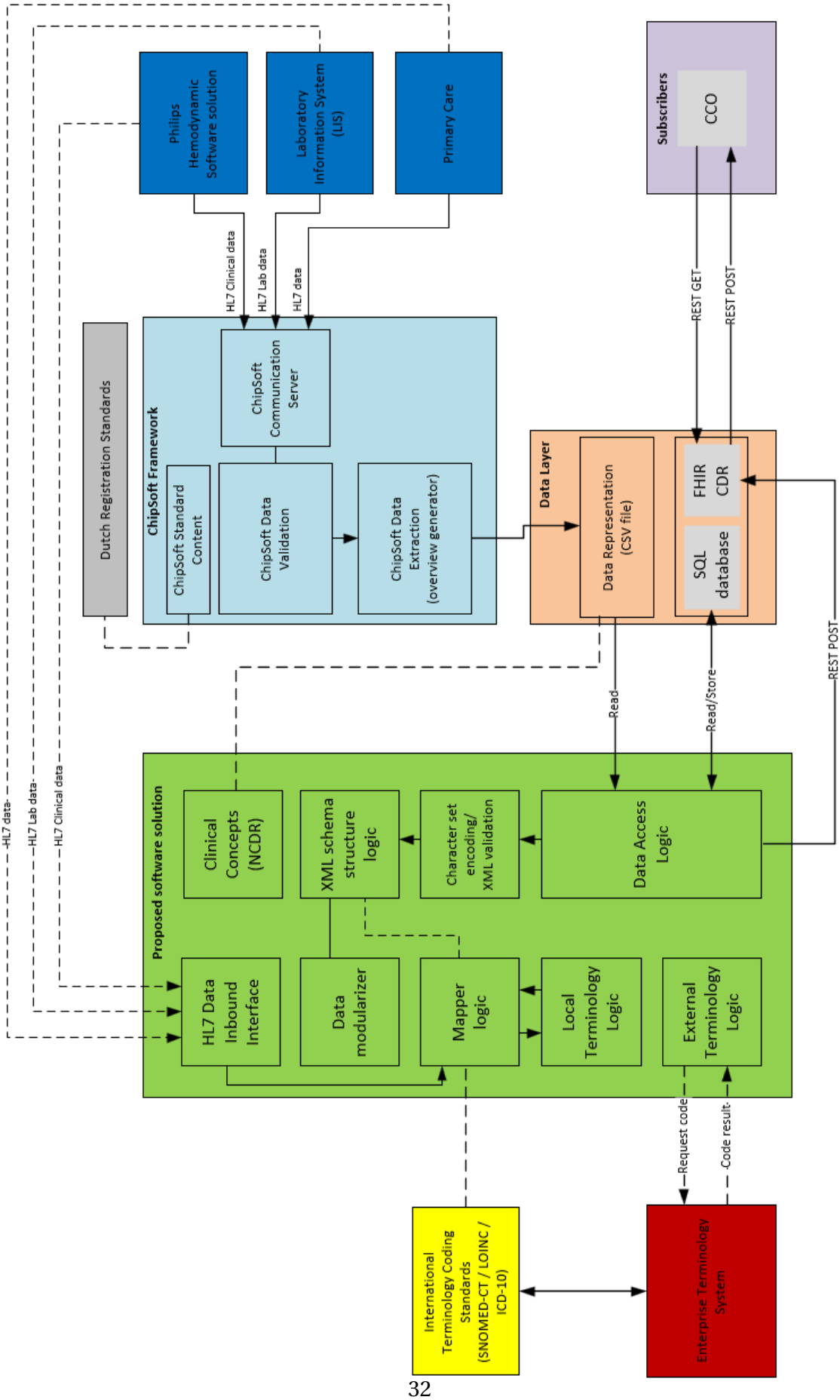

Figure 13: Schematic overview of the general design concept

warehouse demands that the data extraction process produces normalized data exchange formats making the solution multi-institutional deployable [Gavrilov et al., 2020]. The lack of standardization between different data warehouse solutions in various clinical environments makes it difficult for a data warehouse to deal with existing EHR variations. Additionally, this data warehouse solution would only be useful if the hospital owns a data warehouse solution. This possible non-existence causes limitations in the solution's scalability and justifies our choice not to use a data warehouse to extract the required information.

Because we focus this research on the ChipSoft framework, depicted light blue in figure 13, we aim to find a ChipSoft specific solution to ensure efficient multi-institution deployability. After a thorough analysis of the possibilities within the ChipSoft platform, we concluded that ChipSoft has a data extraction tool available (called overview generator) within the software platform able to generate data overviews on any possible data channel. The content extracted from the EHR using the overview generator constitutes a representation of the information as registered within the ChipSoft EHR platform. The reason for choosing the ChipSoft overview generator is to make data extraction queries easily interchangeable between different hospitals. This opportunity enables us to query diverse ChipSoft EHR implementations through a predefined query aligned with our software solution.

6.2. Data Representation

Although we focus on the EHR software platform, acting as a single source of truth to extract clinical data, EHR systems can represent clinical information differently. Despite various initiatives to promote standardization in healthcare, standardization lacks between different multi-vendor EHR implementations. Analysis shows that EHR vendors themselves determine how they represent data internally in the EHR platform. Since EHR vendors also see the need for standardization to introduce uniformity in healthcare, ChipSoft promotes standardization by the unambiguous registration of clinical concepts using standard content. We can define standard content as a ChipSoft-specific coding system containing predefined clinical content and guidelines of different clinical concepts to promote standardized information exchange across medical organizations using the ChipSoft EHR. By employing standardized representations, it is possible to request data in a standardized manner for the ChipSoft framework. Standard content can offer uniformity promoting data sharing capabilities, and ensures more standardization during cardiovascular reporting within the EHR. The ability to identify unambiguously clinical concepts across various medical institutions is fundamental for our research. Using standard content will be essential to deliver a software solution deployable among diverse clinical settings. Future research should determine whether we can apply similar extraction techniques for other EHR vendors.

Using Chipsofts' overview generator, we emphasize the use of standard content to make optimal use of standardization. The extraction method allows us to preload a controlled vocabulary for specific clinical concepts into our practical implementation. Doing so streamlines data capture across different hospitals and returns at any time a controlled vocabulary dataset. The orange data representation box, depicted in the data layer of figure 13, represents a controlled vocabulary dataset returned by the ChipSoft extraction tool. We

foresee that the query results will depend heavily on how clinical practitioners register the data in other medical institutions. The applied clinical workflow and clinical practitioners' willingness play a crucial role in registering clinical data. Therefore, clinicians must adapt their working methods to register data unambiguously and efficiently. We will validate our query's reusability in various hospitals during section 8.2 where we handle scalability validation.

As described above, we can rely on ChipSoft's overview generator tool to accomplish the first step towards an unambiguous and consistent data extraction method. Using a generic query, we can extract specific cardiovascular concepts from the cardiovascular report defined in the ChipSoft software platform. The defined cardiovascular report includes all relevant parameters recorded before, during, and after the patient's PCI procedure within the ChipSoft framework. Since ZOL Genk acts as a ChipSoft reference site in Belgium, participating in the cardiovascular report development, all clinical concepts embodied in the reference report are based on ChipSoft standard content. On the other hand, the ChipSoft overview generator allows us to transfer predefined queries between different ChipSoft implementations returning clinical concepts defined as standard content. This approach empowers us to apply the data extraction method in a multi-institutional environment. The query's reusability allows us to generate an output aligned with our proposed software solution. Building this query required an enormous amount of time and knowledge about the EHR's data structures and location of the cardiovascular concepts. We elaborate further on practicing the data extraction solution provided by ChipSoft in section 7.1.

A potential challenge for data interoperability within healthcare is how EHR vendors and clinical practitioners apply proprietary clinical terminologies. Because clinical physicians are hugely involved in these software platforms' daily use, they can significantly improve the usability of the software [Reisman, 2017]. On the other hand, they introduce data variations between different healthcare organizations. Since other healthcare institutions sometimes see the usefulness of certain local adjustments, ChipSoft integrates regularly customized hospital demands with proven service into the standard content and identifies these concepts with an internal Chipsoft standard content identifier. Subsequently, a Chipsoft managed distribution mechanism ensures the transfer of this newly added entity to all hospitals using the ChipSoft EHR, bringing uniformity between diverse clinical environments.

Figure 13 illustrates a general schematic overview where various clinical data repositories contain important cardiovascular information. We represent these CDRs as dark blue entities. As discussed earlier, we assume the centrally deployed EHR platform integrates cardiovascular information originating from diverse CDRs through HL7 v2 communication. ChipSoft uses a universal communication server responsible for all HL7 v2 data integrations with the EHR software platform. Integration consultants must ensure data conversion validity according to ChipSoft's clinical content definition before accepting data from clinical practitioners and suppliers' software systems. The light blue depicted 'Chipsoft Data Validation' entity ensures the validation of incoming data within the ChipSoft framework. Because we assume that not all CDRs integrate their valuable cardiovascular information with the EHR platform, we offer a second extraction mechanism to capture

cardiovascular information. On top of the extraction mechanism to capture cardiovascular information from the EHR platform, we provide a second extraction mechanism to collect cardiovascular information directly from HL7 v2 data. This second extraction mechanism is handled by the 'HL7 Data Inbound Interface' software component in the green depicted proposed software solution entity in figure 13.
We can now make the previously discussed interoperability solution, represented in figure 7, more specific by unambiguously representing cardiovascular information to a common data model in two ways. The proposed prototype offers the possibility to capture cardiovascular information from two data extraction channels. A first extraction channel provides a semantic mapping towards a common data-model by capturing HL7 messages, while a second extraction channel provides this semantic mapping from multi-vendor EHR systems. In addition to the current implementation delivering data integration from the ChipSoft EHR and the HL7 data channel, we can scale-up the prototype to support more vendor-specific EHR extraction channels. All realized integration channels of the prototype we marked green in figure 14.

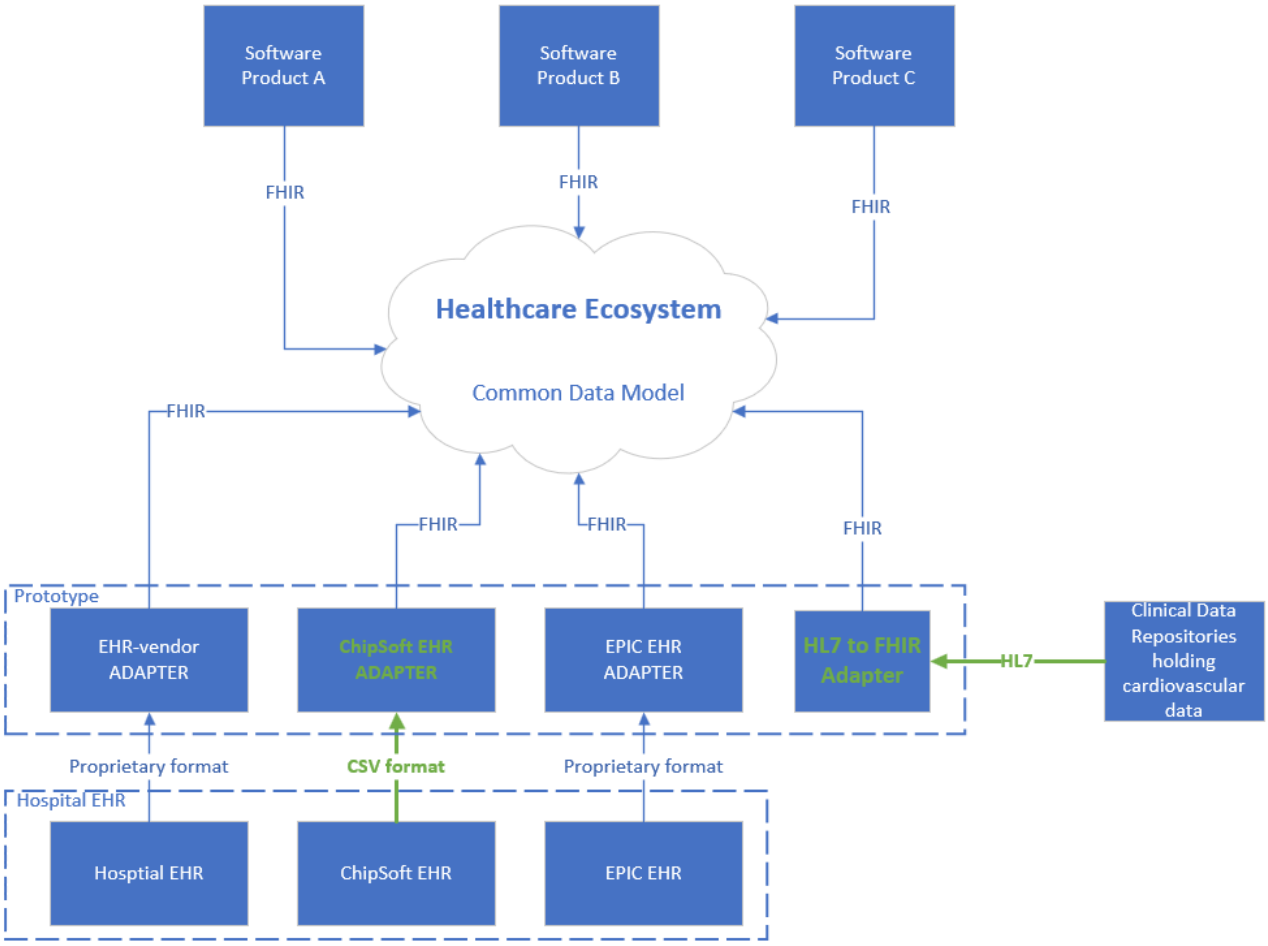

Figure 14: Extended data model

6.3. Data Transformation

Data transformation is one of the biggest challenges and is the most complex part of the ETL process [Ong et al., 2017]. To develop a multi-institutional solution suitable to express clinical information in a standardized way, we need to represent the clinical information as much as possible towards internationally recognized terminology systems. We need to achieve this by delivering a syntactic and semantically harmonized representation of the source dataset after performing a series of operations on the source dataset to comply with the target clinical data model. We represent this series of operations as a data transformation. Data transformation is converting the data and structure of a source data element into a required data format of the target system [Mate et al., 2015].

Since we want to strengthen our solution, green depicted in figure 13 with debugging and troubleshooting capabilities, we aim to store the raw data (CSV-file) untouched in a database. Continuously having access to raw data enables us to identify and resolve issues faster. This mechanism, managed by the software component 'Data Access Logic', offers the possibility to control and manipulate the transformation process during development activities from the SQL database environment. Once the source data is in place and extracted by the 'Data Access Logic' from the SQL environment, we want to tackle data quality as soon as possible. This pre-transformation phase, depicted in the green area as the 'Character set encoding/XML validation' entity, must make the data suitable for further transformation by the proposed solution. Since one of the main tasks of a data transformation is about reformatting data structures, we want to describe the data representation in our software solution. This description defines the XML message structure the message must conform to before processing by the integration engine. We depicted the message definition logic into the 'XML schema structure logic' entity of figure 13. Describing data structures is essential as we want to reformat data structures towards different hierarchical output formats. Establishing such a reformatting between two data structures is achieved through a mapping file incorporating both messages' input and output XML schema structure. Because the XML schema structure and the 'mapper logic' entity are highly dependent on each other to fulfill data transformation, we connected in figure 13 both entities with a dotted line in the proposed software solution artifact. To efficiently execute and manage the transformation process of the obtained data from Chipsofts' overview generator, we aim to modularize this process into different sub-processes. ETL modularization is a technique to abstract common tasks into reusable units of work. We want to modularize the ETL process by recognizing patterns in the obtained dataset Simitsis et al. [2009] to model extracted source data concepts according to their corresponding FHIR resource. Consequently, based on the pattern analysis, we realized direct mappings between the source data concepts and their associated FHIR resources. One of the significant advantages of applying a modulated design is splitting the transformation process into different sub-processes. By separating the main transformation process into diverse (parameterized) sub-processes, the software solution becomes more transparent and better prepared for potential future extensions.

To maintain a clear overview of all clinical concepts during the ETL process, we give each extracted clinical concept a unique identifier. Section 5 described earlier that this research refers to a cathPCI dataset published by the NCDR. This dataset refers to various clinical concepts relevant to cardiovascular PCI procedures suitable to the American healthcare

quality monitoring program. Because the PCI dataset uses a wealth of standardized data concepts, it represents each clinical data concept by a unique numeric identifier. Figure 15 represents an example of such a clinical concept expressing a 'Percutaneous Coronary Intervention Indication', presented as concept '7825'. We have applied a similar methodology within our software solution to identify each clinical concept as a numerical data element. This method helps us grouping all extracted clinical concepts into predefined categories based on the identified numerical data element.

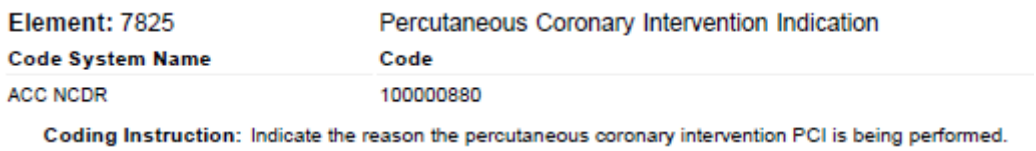

Figure 15: Reference towards clinical concept

By establishing a linking mechanism to link clinical concepts to a specific modularization process, we aim to prepare our solution to support future extensions. This linking mechanism ensures a relatively straightforward modification of the solution without changing the code. To manage data transformations, we mainly use lookup tables in our software solution. A lookup table is a multi-column data table containing input values to provide a simple code-mapping solution without a database. The advantage of such a solution is that lookup tables offer a centralized solution to manage all code mappings in the integration solution. Additionally, lookup tables inject input validation as they only return translated values in case source information is adequately formed. In the case of no matching values, rejection of the input occurs. Although the FHIR specification also provides options to manage relationships between diverse clinical concepts by implementing the ConceptMap[28] FHIR resource, this approach is more challenging in terms of maintainability because integration consultants cannot efficiently perform modifications. Lookup tables offer a much better solution because they are better accessible without changing the code and significantly affect the solution's maintainability and implementation time.

7. Solution Implementation

This section elaborates on the entire prototype development process we have realized in close collaboration with ZOL Genk to reveal the practical feasibility and shortcomings specific to the ChipSoft framework. In the subsections below, we discuss the entire ETL process shaping the entire software architecture based on the software design concept discussed in the previous section. Figure 26 depicts the complete solution architecture diagram. Since we have not yet answered entirely the previously defined sub-research question three [SRQ3b], we will first answer SRQ3b based on the practical experience gained during the extraction of clinical information from the ChipSoft EHR. Subsequently, we aim to answer sub-research question four [SRQ4] on the results obtained by SRQ3b. To answer this question, we developed a prototype to determine how we can realize the data transformation towards an interpretable clinical dataset.

[28] https://www.hl7.org/fhir/conceptmap.html

7.1. Data Analystics

In this subsection, we first answer sub-research question 3b [SRQ3b] and subsequently discuss how we can practically extract information using the ChipSoft generator. Subsequently, we analyze the extracted cardiovascular ChipSoft dataset and model the clinical concepts towards their corresponding FHIR resources. We will then analyze these data modeling activities to optimize the prototype's transformation process based on pattern recognition.

Since we based this study on the clinical concepts specified in the CathPCI data dictionary, defined by the NCDR, we experienced difficulties during implementation to find all defined clinical concepts within the ChipSoft EHR. One of the key limitations we encountered relates to unclear clinical concept definitions included in the cathPCI registry data set compared to the clinical concepts defined in the EHR [Faxon and Burgess, 2016]. A reasonable cause of this is the existence of different quality registration entities operating in diverse countries and the lack of standardized clinical concepts across registries and EHR systems [Flynn et al., 2005]. Consequently, depending on the geographic region, EHR systems record clinical information differently and do not always capture clinical concepts as defined by data quality registration entities. For example, we cannot find a reference in the ChipSoft EHR to the clinical concept 'patient origin' while it is part of the dataset defined by the NCDR. The reason is that Belgian and Dutch PCI quality-oriented registries are not asking for this information. Additionally, the Covid pandemic prevented us from extracting even more complex clinical concepts and medication data from the EHR, for which we required the assistance of cardiologists. Ultimately, we were able to identify 67 clinical concepts in a structured format from the EHR out of a total of 345 defined clinical concepts, representing approximately 20% of the complete cathPCI dataset. With the support of cardiologists, we envision expanding the currently extracted dataset by 80%, representing approximately 35% of the total number of clinical concepts defined in the cathPCI dataset. Due to the ever-increasing collaboration between medical institutions and the drive for standardization in healthcare, we expect an accelerating trend using structured data in the coming years. This trend may influence our results positively and make the solution even more attractive in the future. Some essential aspects to justify this trend are the lower efforts and costs required to ensure structured data quality compared to those required when using unstructured data [Galetsi et al., 2019].

Figure 16 represents an example query created by the ChipSoft overview generator to extract the clinical concept 'Stress Test Performed'. As mentioned earlier in section 6.3, this clinical data concept is referencing to data element '5200' of the PCI data-dictionary. The query's output returns a boolean value stored into the data element Stress_Test_Performed5200 delivered to our solution. First, the query refers to a unique collection (CS00396378COLL) object representing various tests performed for the PCI study. ChipSoft identifies each ID, starting with 'CS' as part of ChipSofts' standard content, present within every ChipSoft implementation in Belgium and the Netherlands. The collection represents a selectable value set for the cardiologists in the EHR software. Next, we filtered the collection 'CS00396378COLL' to remain only the performed tests representing a stress test. Depending on the number of selected tests representing a stress test, the query assigns a boolean value indicating the query's outcome. A boolean value ultimately shows whether or not a stress test took place

during the PCI examination. In the appendix of this study, we list three different examples part of the entire ChipSoft query. The query results represent a consistent and fully implementable dataset aligned with the proposed software solution, interchangeable between different ChipSoft EHR implementations. Figure 40 shows a first example where it is possible to access the patient's first name directly. This type of query is the easiest method to access data via the ChipSoft generator tool. A second example illustrates in figure 41 a query where we applied a simple expression to determine whether the patient is a tobacco user or not. As a final example, we present in figure 42 a more advanced query to determine the application of specific cardiovascular tests during a PCI procedure. In those figures, we highlighted the query expression in yellow and the corresponding clinical concept containing the query results in green.

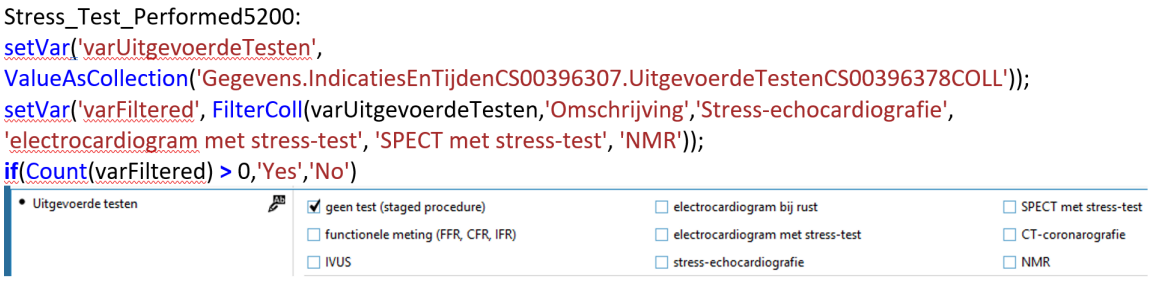

Figure 16: Example of data extraction using ChipSofts' overview generator

After running the developed query in the ChipSoft extraction tool, we obtain a comma-separated values (CSV) file holding all the extracted information. Suppose we reflect this action on the entire ETL process. In that case, we can associate the earlier described data preparation stage, aligning the cardiovascular information with our prototype, to the extraction process as the first step towards an unambiguously defined dataset. The query's strength envisions the interchangeability between various healthcare institutions that implemented ChipSoft as an EHR. Next, our software solution needs to pick up the returned CSV file containing all extracted data from the EHR for further processing.

Once the CSV input file is in place, questions arise about the second step in the ETL process, transforming the CSV input towards the corresponding FHIR resources. As the representation of clinical concepts into CDRs is subject to variation, we need to consider a flexible mechanism to translate various vocabularies towards internationally recognized terminologies. For this reason, we aligned the ChipSoft data transformation mechanism entirely with the integration solution's existing translation mechanism handling the semantic mapping of the HL7 v2 data channel. This architecture's advantage empowers us to manage diverse extraction channels' semantic mappings through one centralized configuration setting managing data transformation. First, we analyzed all extracted clinical data concepts embodied in the output of the ChipSoft overview generator. Using the FHIR specifications, published on the HL7 FHIR specification website[29], we linked each extracted clinical concept towards their corresponding FHIR resource. The FHIR standard defines a FHIR resource as a blueprint representing a building block to exchange healthcare data.

[29]http://hl7.org/fhir/

To maintain a clear overview while mapping all extracted clinical concepts to their corresponding FHIR resources, we used a mapping worksheet, an Excel file supporting us to understand better the data representation requirements conform to the FHIR standard. This approach has made it possible to model all data elements in associated FHIR resources and enrich them semantically with their corresponding codings. Figure 19 represents a schematic overview in UML style, providing an overview of the applied FHIR resources after mapping all obtained clinical data concepts. The data model depicts a graphical representation of the required FHIR resources and their dependencies, expressing a general data model needed to convert all obtained cardiovascular data concepts from the ChipSoft framework to FHIR resources. Table 1 in the appendix of this study presents a brief description of the meaning of all used FHIR resources. Figure 17 shows a graphical overview reporting the variety among the number of FHIR resources required to model the extracted cardiovascular information originating from the ChipSoft EHR. Based on a total of 67 modeled FHIR resources, we determined that we can model most clinical concepts, 52 to be exact, as FHIR Observations resources. In general, FHIR Observation resources represent measurements and assertions made about a patient undergoing an examination in a cardiac intervention center.

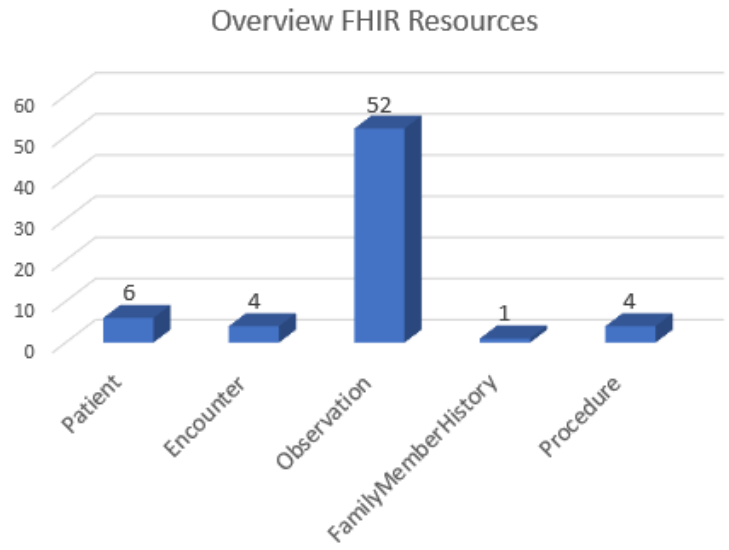

Figure 17: Diversity among the number of applied FHIR resources required to model ChipSoft input

When modeling a FHIR Observation resource, we can determine the observation type based on the observation result. In the previous section 6.3, we described this process as data modularization. A simple example is the modeling of an observation in which the answer can be yes or no. Such kinds of observation types we modeled as valueBoolean observations representing a boolean output. In addition to the valueBoolean observation types, we recognized other observation types to model specific obtained information correctly. Figure 18 represents a subdivision of the 52 classified FHIR observation resources based on the various recognized observation types. In section 8.2, validating the prototypes' scalability, we discuss in more detail the numbers shown in figure 18. We aligned our solution with all determined FHIR Observation resource types by creating sub-functions responsible for

mapping these appropriate observation types. As discussed earlier, this modularization technique improves the code's readability and maintainability [Sedano, 2016]. Adjustments considered necessary for specific observations can be much easier to locate than handling all mapper activities into one large and unclear main mapper.

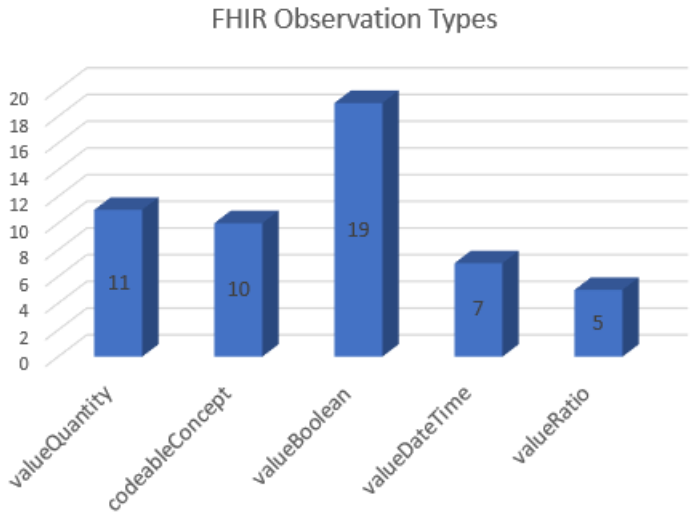

Figure 18: Diversity among the FHIR Observation types

Furthermore, due to the growing volume and variety of data within an EHR, we also have to deal correctly with the concept of data quality. Data quality in healthcare must consider several characteristics, including accuracy, validity, and consistency [Feder, 2018]. Data accuracy is an essential feature for exchanging clinical data and clinical decision support systems [Van Hoeven et al., 2017]. By applying validation techniques before accepting clinical data from diverse CDRs, EHR solutions validate patient information and several clinical concepts. As we retrieve the information directly from the EHR, our software solution can immediately perform the necessary data transformations to correctly model various extracted clinical measurements towards their corresponding FHIR resource. The concept of data consistency is a fundamental property for our software solution, as we want to store clinical data in a standardized and unambiguous way. Returned clinical concepts from the EHR, implemented in different medical settings, must be predictable to enable data transformations towards unambiguously defined clinical concepts. The presence and use of standard content in the Chipsoft EHR are crucial for delivering data consistency among diverse clinical environments. According to Aerts [2018], it is preferable to avoid data transformations as much as possible during mapping activities to ensure data quality. Any form of data transformation opens up the possibility of introducing errors and reducing the data's reliability. By making optimal use of the ChipSoft standard content in developing the extraction query, we strongly promote the returned values' predictability across multi-institutional environments. In section 7.3 we describe in more detail how we can subsequently flexibly manipulate this returned data to facilitate conversion to the desired output.

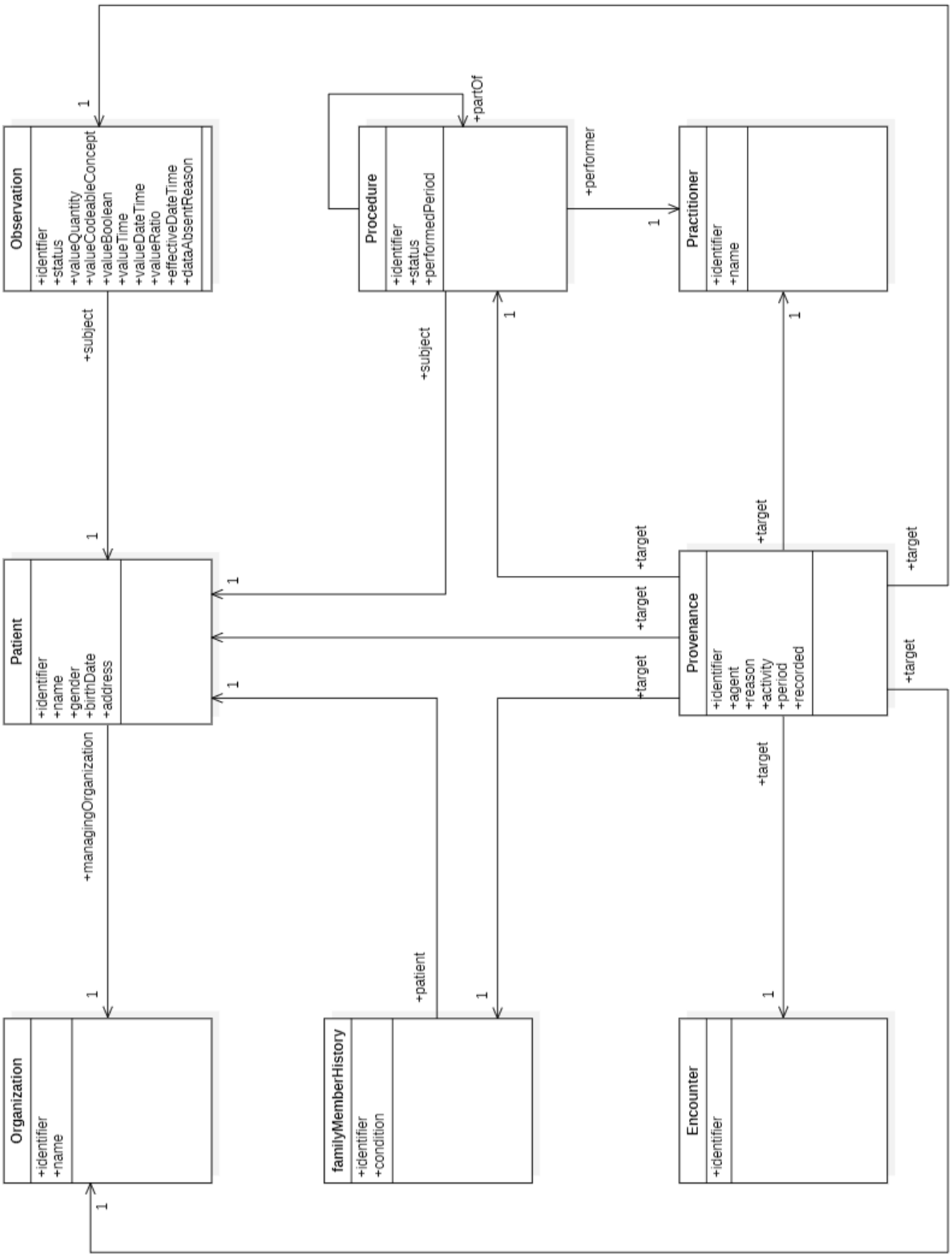

Figure 19: FHIR datamodel

7.2. ChipSoft Data Import

This subsection explains how we will practically import the CSV content extracted from the ChipSoft EHR into the prototype. As discussed earlier in subsection 4.3, we decided to build our software solution on the Orion Health Rhapsody platform to maintain interoperability with other Philips interoperability solutions. Central to the concept of this integration platform are routes and communication points (CPs). Routes form the path taken by the message passing through the Rhapsody integration engine, while CPs connect with those paths. Throughout the design process of our software architecture, we employed design patterns with a proven integration solution efficiency to minimize the complexity of the software architecture's ability to handle variations in a flexible manner [Hohpe and Woolf, 2004]. Further, we want to maintain a clear overview of the entire message flow between the various software components. The first developed route available in the software architecture diagram, represented in figure 20, handles data import.

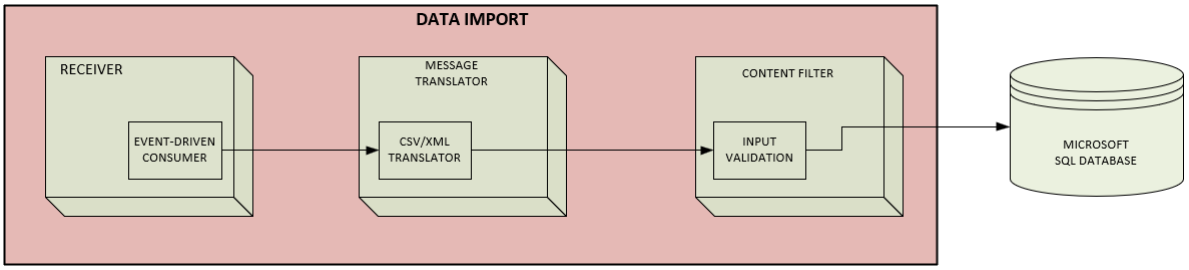

Figure 20: Data import route

Once the ChipSoft extraction tool drops a CSV file, the CP of the data import route is triggered to pick up the file, starting processing the content. The intention is that the ChipSoft extraction tool systematically delivers a CSV file to keep the FHIR CDR information up to date. An event-driven daily scheduled task, triggered by the ChipSoft task server, can systematically deliver new cardiovascular data extractions from the ChipSoft EHR. Since cardiologists regularly report their conclusions about the cardiovascular examination later than the examination date, minor adjustments to the cardiovascular report may occur after the surgery date. For this reason, we propose that the daily data extraction includes the content of all cardiology examinations performed in the past seven days.
Because Rhapsody's internal data format is XML, the route handles the message translation of the CSV content to an XML structure. A script written in Javascript (CSV/XML translator process) takes care of this, verifying the input of the CSV content.
The input validation process prevents specific input from causing the XML code to be found syntactically incorrect [Grijzenhout and Marx, 2013]. It is also an essential validator to prevent fraudulent practices by injecting malicious code.
Finally, this route stores the data in a Microsoft SQL relational database. Furthermore, the data import route's output CP contains logic to check the input content's existence in a SQL database. The output CP verifies the cardiovascular report's presence into the database based on a unique identification number referring to the cardiovascular report extraction. The outbound CP creates a new SQL entry representing a cardiovascular report extraction in the case of non-existence. In contrast, the existence of a cardiovascular extraction report triggers an update of an existing SQL entry. We identify non-existing and updated entries in the database with a processing flag set to zero to indicate the SQL database's cardiovascu-

lar report's processing state. Determining whether to process specific SQL entities further depends on the value of the processing state.

7.3. ChipSoft Extraction Framework

After converting the CSV file content to a SQL table, the solution is ready to convert the cardiovascular reports towards their corresponding FHIR resources. It is important to note that the research's elaborated solution has mainly focused on data processing obtained from a ChipSoft EHR extraction. To minimize the software architectures' complexity, we want to decouple the various EHR software platforms from the prototype. On the other hand, loose coupling requires maintaining a clear overview of the message flow. To keep track of the message flow in the prototype, we want to label every incoming message. Message labeling can help the software solution manage message flow routing by providing each particular source CDR (EHR platform) a vendor-dependent label specifying the extracted data's origin. Based on these labels, the software solution can apply specific routing decisions to facilitate different EHR data representations.

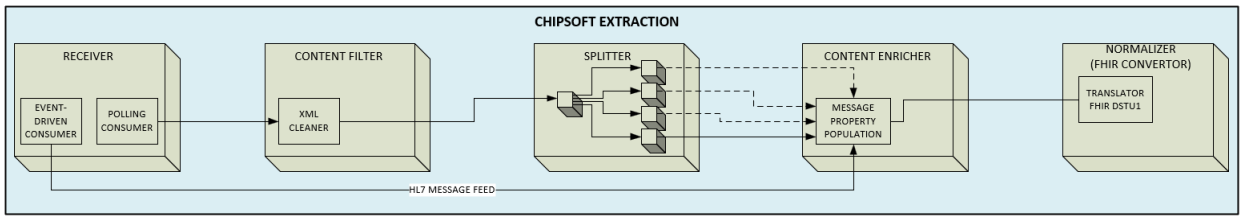

Figure 21: ChipSoft extraction route

Figure 21, displays a polling consumer CP scanning unprocessed entries in the SQL database. This polling mechanism is checking each second for unprocessed entries. The polling consumer is a database CP in input mode, used in Rhapsody to poll a SQL table to detect changes. Once the polling mechanism detects an unprocessed entry, the CP executes a customized query, generating XML data representing the extracted data. It is essential to mention that HL7 v2 data originating from diverse CDRs can also enter this processing route through the event-driven consumer CP. After converting all messages to XML, the extraction route will merge all messages to fulfill a semantic mapping of both data sources in one place. This design delivers a significant advantage in terms of maintainability since we offer an interoperability solution to manage all semantic mappings in one central place, no matter how many EHR systems the prototype supports.
An analysis of the XML structure after the polling consumer (EHR channel) reveals that we could improve the polling mechanism's standard XML format. The default applied XML schema after the polling process displays the data in an unstructured style. Since we are looking for a *scalable* and *performant* solution, we prefer to obtain a suitable XML structure immediately after running the custom query without manipulating the XML layout. A well-thought-out XML structure must enable us to deal with data transformations efficiently. By immediately obtaining the XML structure in the desired format, grouped by the FHIR resources obtained from SRQ3b, we want to facilitate iteration to make the transformation process more efficient and *scalable*. Introducing an iterative character in the XML structure should enable the solution to support adding new FHIR observations effortless

without making changes to the code. For example, extending the prototype with a clinical concept categorized as a particular FHIR resource is just a matter of incorporating the data within the XML hierarchy's desired element. Figure 22 shows part of the query applied to define the XML hierarchy of the extracted data without manipulating the default XML format generated by the integration platform.

```
SELECT top(10) Extraction2.ID AS cathlabID,

CAST ( (SELECT
'UniqueID' AS "Cathlabreport/Code", Extraction.[ID] AS "Cathlabreport/Value",

--  FHIR Patient resource mapping
'2000' AS "Patient/Last_Name/Code", Extraction.[Last_Name2000] AS "Patient/Last_Name/Value",
'2010' AS "Patient/First_Name/Code", Extraction.[First_Name2010] AS "Patient/First_Name/Value",
-- ...

-- FHIR Encounter resource mapping
'3001' AS "Encounter/Arrival_Date_Time/Code", Extraction.[Arrival_Date_Time3001] AS "Encounter/Arrival_Date_Time/Value",
'3050' AS "Encounter/Admitting_Providers_Last_Name/Code", Extraction.[Admitting_Providers_Name3050] AS "Encounter/Admitting_Providers_Last_Name/Value",
-- ...

-- FHIR Observation resource mapping
'4615' AS "Observations/Observation/Code", 'Hypertension' AS "Observations/Observation/Name", Extraction.[Hypertension4615] AS "Observations/Observation/Value','' AS "Observations",
'4620' AS "Observations/Observation/Code", 'Dyslipidemia' AS "Observations/Observation/Name", Extraction.[Dyslipidemia4620] AS "Observations/Observation/Value','' AS "Observations",
-- ...

-- FHIR FamilyMemberHistory resource mapping
'4287' AS "FamilyMemberHistory/Family_Hx_of_Premature_CAD/Code", Extraction.[Family_Hx_of_Premature_CAD4287] AS "FamilyMemberHistory/Family_Hx_of_Premature_CAD/Value",

-- FHIR Procedure resource mapping
'7000' AS "Procedure/Procedure_Start_Date_Time/Code", Extraction.[Procedure_Start_Date_Time7000] AS "Procedure/Procedure_Start_Date_Time/Value",
'7005' AS "Procedure/Procedure_End_Date_Time/Code", Extraction.[Procedure_End_Date_Time7005] AS "Procedure/Procedure_End_Date_Time/Value",
-- ...

FROM CS_Extraction Extraction
where Extraction.[ID] = Extraction2.ID
FOR XML PATH ('Extraction'), ROOT ('ChipSoft'), ELEMENTS XSINIL) AS VARCHAR(MAX)) AS XmlData
from CS_Extraction Extraction2
where Extraction2.processed = 0
```

Figure 22: Part of ChipSoft Polling query

In addition to scalability, we also consider *performance*, the time needed to fulfill the entire ETL process, as an essential requirement of the software solution. The query's design impacts the solution's performance as we can avoid an extra mapper by immediately obtaining the XML structure in the desired format. Since mappers require an investment in time and resources, we can gain significant time savings by eliminating an additional mapper to get the data into the required format. In section 8.3, we describe performance validation in more detail. Another essential element accelerating the performance of our solution is the processing of messages in batches. This technique is also used in other domains and ensures merging several messages as one message [Avilés-González et al., 2016]. This batch operation aims to drastically reduce overhead by eliminating the metadata produced from each message.
The content filter component removes the metadata from the extracted data and outputs only refined extracted information.
The next applied design pattern used within this route is the splitter pattern. We apply this design pattern to divide a message consisting of several elements into separate messages. We call this process de-batching.
Subsequently, the message enricher design pattern enriches each separate message obtained after the splitter pattern with additional content. The extra information added to the message is irrelevant and employed within the software solution to distinguish between the messages' origin. The software solution assigns a distinction label based on a source identification label. Using the source identification label, we can later make specific routing decisions based on the message's origin. The message content enricher permits the solution to route messages from various EHR systems to separate data mappers. The Rhapsody

variable manager provides a mechanism for specifying environment-specific parameters to configure properties for related components. This technique introduces enormous flexibility within the software solution to make appropriate choices without adjustments within the code. Once the content enricher process assigned properties to the XML messages to control the message stream, the solution is ready to accomplish the extracted data mappings to the corresponding FHIR resources' elements.
Since FHIR is a continuously evolving standard Benson [2010], we need to prepare our software architecture to handle distinct FHIR versions efficiently. To develop a solution that can flexibly deal with this variety of FHIR standards, we offer a solution with two data translators. The primary data translator, depicted in figure 21 is responsible for translating the extracted content into a normalized version. A second mapper discussed later in subsection 7.4 manages the normalized FHIR version's translation into the desired FHIR standard.

The existence of a data integration solution to convert HL7 v2 data channels to FHIR resources allowed us to develop further and fine-tune an already existing software architecture. Moreover, we could reuse existing message definition files applied in Rhapsody describing various FHIR resources' structure. To speed up the prototype's development process, we developed our software solution based on earlier Philips's design decisions to manage the variety in FHIR versioning. To build a flexible conversion mechanism to manage diverse and upcoming FHIR standards, software architects decided to use an older DSTU1 standard as a reference standard during the FHIR toolkit development process. Due to this decision and the ability to reuse the existing message definition file describing the FHIR DSTU1 message structure, we used a message translator pattern to map the extracted cardiovascular content towards the existing referencing (normalized) FHIR specification standard. Before creating a Rhapsody translation component ('translator FHIR DSTU1' component in figure 21), we must define two message definitions describing the input and the output message structure. Regarding the message translator's input, the input message's definition depends on Rhapsody's incoming XML structure after processing the offered CSV content. In contrast, the message translator's output definition file is reusable from the already existing second message translator described in the following subsection 7.4 as the second message definition's input definition is similar to the output definition of the first message translator component. Once both message definition files are in place, a Java mapping engine manages the necessary mapping activities. Such a mapping engine consists of a primary function with two parameters referring to the mapper's input and output message. To achieve this, we implemented separate sub-functions for each unique FHIR resource holding the logic to map all corresponding FHIR resource data elements. As discussed earlier, delegating the logic to sub-functions empowers us to develop and maintain the code easily. Figure 43 shows an example where the mainChipSoft function delegates responsibility for each FHIR resource mapping towards a corresponding sub-process.
To realize consistent translations, the RhapsodyTableLookup() function, depicted in listing 1, permits us to build flexible lookup operations by enabling the data translator to look up specific data entities and translate them accordingly. Figure 23 depicts an example of the lookup entry managing the translation of the clinical concept extraction with ID '7825'. For this specific observation, the RhapsodyTableLookup function in listing 1 performs a lookup to translate the obtained content towards a terminology code, representing the meaning of

the clinical contents' value. The function RhapsodyTableLookup() provides access to the lookup table 'ChipSoft _NCDR_CodeableConcepts_NormalizedTranslation' parameterized with several parameters listed in figure 44 of the appendix to facilitate a dynamic translation towards a clinical code defined in the cathPCI dataset.

```
RhapsodyTableLookup(gNCDRCode,"
    ChipSoft_NCDR_CodeableConcepts_NormalizedTranslation","NCDR_Code","
    ","ObservationCode",observationCode,"Text_To_Be_Looked_Up",value);
```

Listing 1: RhapsodyTableLookup function translating an observation value for a specific observation code

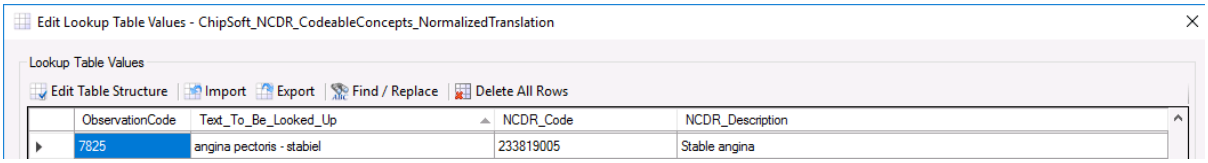

Figure 23: Rhapsody lookup table managing the translation of the observation value.

By applying this translation method, we could represent all clinical concepts unambiguously according to international terminology standards. Because we intensely experience that the mapping of clinical terminologies is not evident, we asked during implementation assistance of a third-party company[30] specialized in clinical concept mappings. Since clinical concepts are not always correctly defined within EHR systems, it is often challenging to find the appropriate terminology defined in international coding systems. The existence of comparable vocabularies in various standards also counteracts the mapping process and introduces confusion. Together with this company, we realized most of the mappings for the ChipSoft extraction. As expected, we also encountered some ambiguities in mapping specific clinical source concepts incorporated into multiple international coding systems. We solved this by linking clinical concepts precisely and with as much overlap in meaning as possible. On the other hand, we also discovered clinical concepts where we could not realize an unambiguous mapping. In this situation, we have applied a fictitious coding standard referring to ChipSofts' standard content.

The effort and complexity involved in establishing correct mappings make maintaining these local mapping tables extremely difficult. Moreover, manual mappings ensure a high error sensitivity of the mappings concept resulting in incorrect conversions. Additionally, we must consider an increase in the number of future mappings to prepare the prototype for a potential expansion supporting multi-vendor interoperability. Consequently, local mapping tables cannot ensure the solution's maintainability to achieve a scalable and enterprise-wide interoperability level. To avoid this shortcoming, we want to delegate the logic of a local terminology solution to a local hosted enterprise-wide terminology system, represented as a red entity in figure 13. This enterprise-wide terminology solution must take responsibility for the necessary translations of different clinical concepts, even for other interoperability solutions provided by Philips. In addition to internationally terminology coding systems, the enterprise-wide terminology solution can also host our already realized mappings of locally applied codes.

[30]https://www.furore.com/

As we strive for a maintainable multi-institutional solution, we propose implementing a centralized terminology solution within the Philips ecosystem. We foresee integrating an external terminology server hosted into the Philips ecosystem through a lightweight and secured REST API call instead of addressing the locally hosted logic. Several sources emphasize the need for server terminology solutions within medical institutions [De Quirós et al., 2018] [Metke-Jimenez et al., 2018]. In the Netherlands, there are very recent developments ongoing in applying a national terminology server that can serve to publish terminologies of national importance to local terminology solutions automatically [31]. This evolution should improve the maintainability of local terminology solutions but lacks the support of local translation tables. Due to the very recent developments, this solution has not yet delivered proof of concept.

To tackle the just discussed problem avoiding confusion when similar concepts exist in different terminology standards, we aim to represent each extracted clinical concept under one common denominator in our prototype. Several research projects discussed in the related work section show that we can solve this problem using UMLS. To integrate UMLS into our software prototype solution, we need to access an external UMLS endpoint through a UMLS API. Since it is not allowed to set up an external API call from our test environment in the hospital with UMLS, we have analyzed how we can implement a practical test scenario. Before demonstrating a practical test, we have applied for a research license to justify UMLS's use within this research framework. Since UMLS requires access to an external environment, and insecure endpoint APIs are vulnerable to security attacks Abrar et al. [2018], we must take the necessary security measures to prevent cyber-attacks. Although security falls outside this study's scope, we hold it is worth mentioning security is priority number one, especially in medical environments [Mattei, 2017]. To set up a secure UMLS call, we first need to request a user access token (ticket-granting ticket) with a limited validity period of 8 hours. After successful authentication, we can request a single-use service ticket (ST). The obtained ST enables us to perform one UMLS call during a time window of 5 minutes. In case we want to implement UMLS within our solution, it is desirable to consider that we first need to set up secure authentication with the UMLS system. We must therefore build in the necessary logic to establish a secure connection before performing a translation. Once we have a secure connection, we can query the external UMLS server to obtain uniform terminology. Based on an internationally recognized terminology concept, we can subsequently obtain a unified view of this previously identified concept by the UMLS standard. Figure 24, illustrates how to translate a SNOMED-CT code, representing a clinical concept 'Stress echocardiography'. In response to this GET request, where we include various URL parameters, the UMLS system returns the corresponding uniform UMLS concept with ID 'C0920208'. When setting up a UMLS call, it is essential to mention that we can provide an equivalence attribute (searchType parameter in figure 24), reflecting the nature of the mapping between the source element and the target element. In case we cannot find an exact match for a particular clinical concept, UMLS empowers us to search for generalized concepts by indicating the degree of equivalence concerning the current clinical concept. To avoid performance problems, we want to minimize the number of external calls and cache realized UMLS mappings locally.

[31] https://www.nictiz.nl/standaardisatie/terminologiecentrum/nationale-terminologieserver/

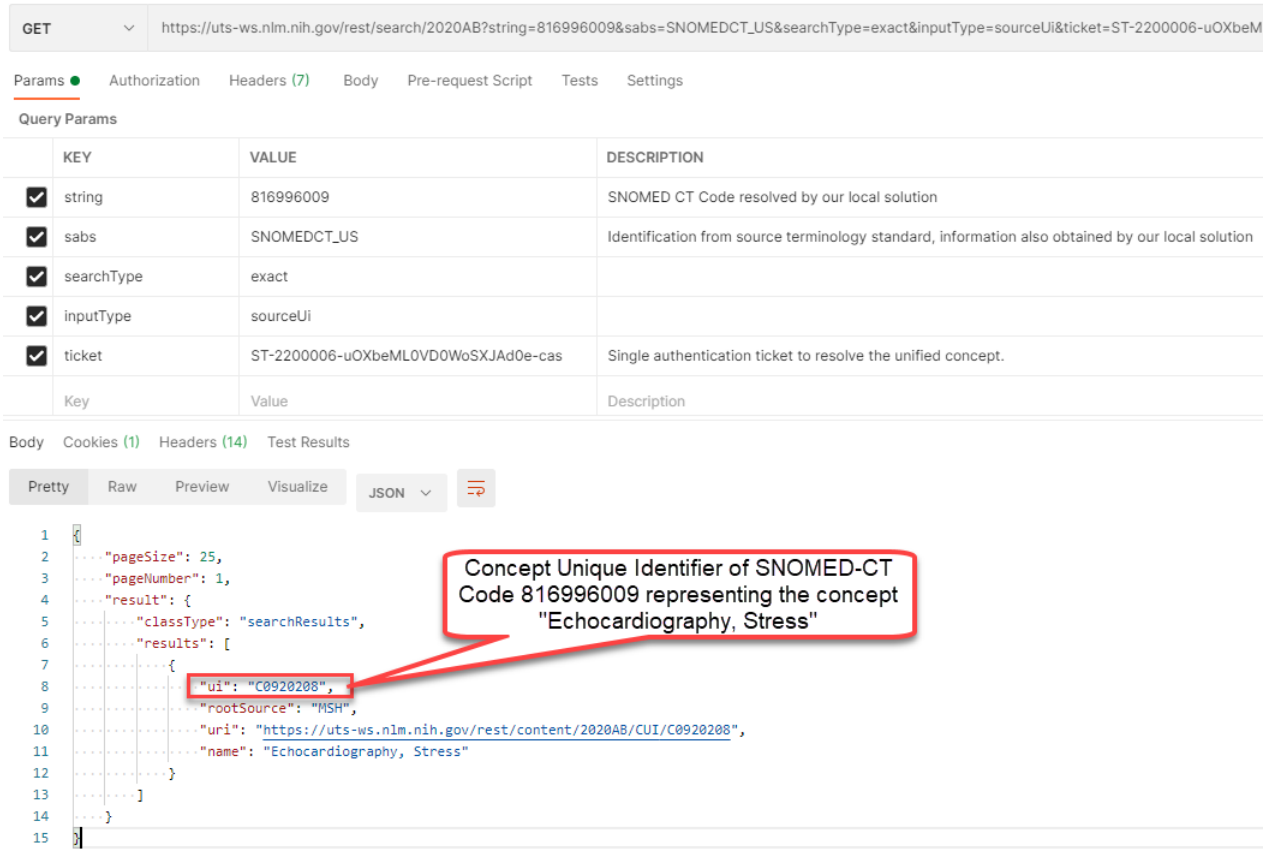

Figure 24: Example of API call to resolve a unified view of a clinical concept

7.4. Data Translator FHIR Toolkit

As mentioned earlier, we have applied a second message translation pattern that permits the solution to handle different FHIR standards dynamically. A second message translator is responsible for mapping the normalized data structure to the desired FHIR standard output. Figure 25 depicts a schematic overview of how our solution handles various standard FHIR output formats.

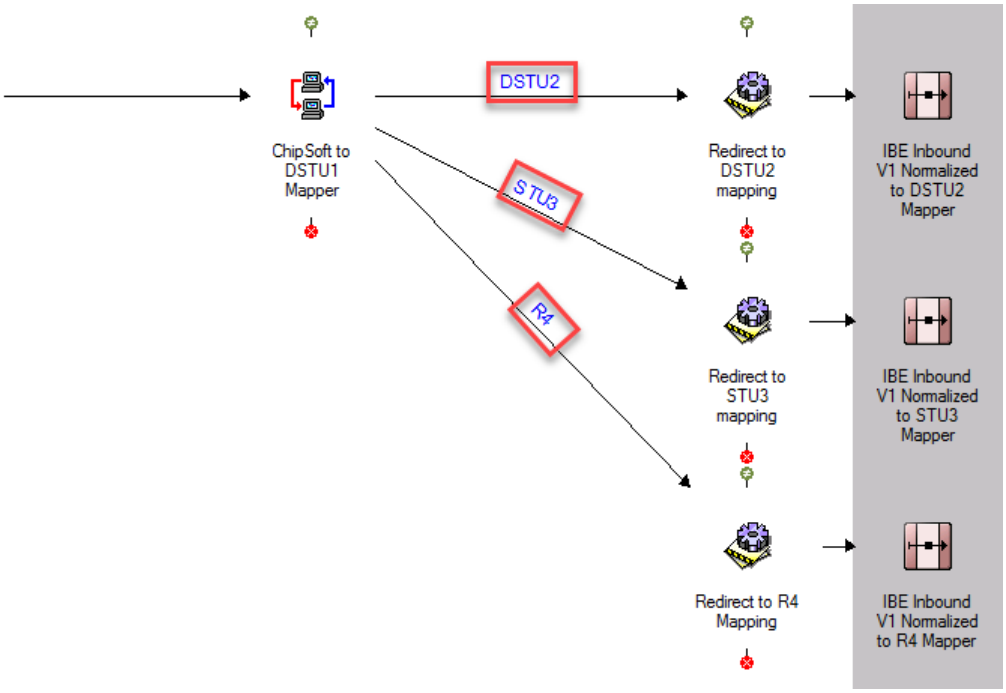

Figure 25: Dynamic Data Translation Mechanism

Because the integration platform makes lookup tables directly accessible from JavaScript, we can easily accommodate this dynamic behavior within the proposed solution at runtime. By labeling incoming messages, we can allow our software to make individual choices during message processing. Conditional connectors, marked by the red boxes in figure 25, selectively route messages to the corresponding message translator based on the configured FHIR version within a lookup table qualified for handling FHIR output type (XML or JSON) and associated FHIR versions. Since we provide each source repository delivering information to the software solution with a source identification number, we can easily manage the output format individually without changing the code. The manipulation of a centrally accessible parameter is sufficient to allow the solution to deal with different standards in various medical settings.
As mentioned earlier, we reused the second mapper code from the existing software solution within Philips. Since we encountered some obstacles during the realization of particular mappings, especially by the lack of specific data elements in the normalized message structure, we had to make minor modifications in the second mapper. All applied changes were well documented in the code using the built-in comment blocks in the respective data translator.

If we observe the data access methods in EHR solutions in the Belgian and Dutch market in practice, we see that global EHR suppliers start offering FHIR APIs [32] [33]. These FHIR APIs can speed up the integration and innovation process between third-party applications and hospital EHR platforms, given that EHR platforms deliver data according to international coding standards. To our knowledge, these direct FHIR integration options to extract cardiovascular information are not available by ChipSoft at the time of writing. However, we do see FHIR APIs emerging for the exchange of *primary* health data between healthcare providers[34]. Due to the gradual adoption of FHIR and the rise of national collaborative care platforms[35], we foresee a decrease in the number of required transformations from HL7 version 2 to FHIR resources in the future.

7.5. Clinical Data Repository

This research aims to collect the various types of clinical information from several clinical data sources unambiguously. In this subsection, we would like to focus on the load process of the ETL process. The clinical data repository is part of the software solution containing all extracted and transformed clinical information. The information is represented unambiguously within this clinical data repository according to an FHIR compliant data model. In figure 45, we illustrate an anonymized FHIR dataset presented by the software prototype published to an FHIR compliant CDR. By representing all clinical information into an FHIR-compliant repository, we can deliver a much more flexible healthcare solution offering a wide variety of development possibilities to build clinical applications.

7.6. Subscribers

Philips is currently working on a state-of-the-art solution to develop a cardiovascular dashboard representing relevant patient-centric parameters to cardiologists at the start of a cardiovascular examination. The proposed architecture allows the ultimate opportunity to build a dashboard based on a FHIR compliant data model. In addition to a cardiovascular dashboard, as illustrated in purple in figures 13 and 26 as the Cardiovascular Care Orchestrator (CCO) entity, the FHIR CDR can provide a solution to many healthcare product solution designs fostering innovation in mobile healthcare [Braunstein, 2018].

8. Solution Validation

We worked out earlier a software architecture through a Design Science Research Methodology and developed a prototype to reflect the proposed software architecture in practice. This section concerns the validation of the proposed software solution, where we want to answer sub-research question five, determining the efficiency of the developed prototype [SRQ5]. We aim to gain a better understanding of the scalability and performance of the software solution. We will first outline the entire validation process and subsequently describe the practical approach to ensure all prototype's published FHIR resources' correct-

[32] https://open.epic.com/Interface/FHIR
[33] http://fhir.cerner.com/
[34] https://www.chipsoft.be/hix-abc/artikel/215/Zorgverlenersportaal
[35] https://www.cozo.be/

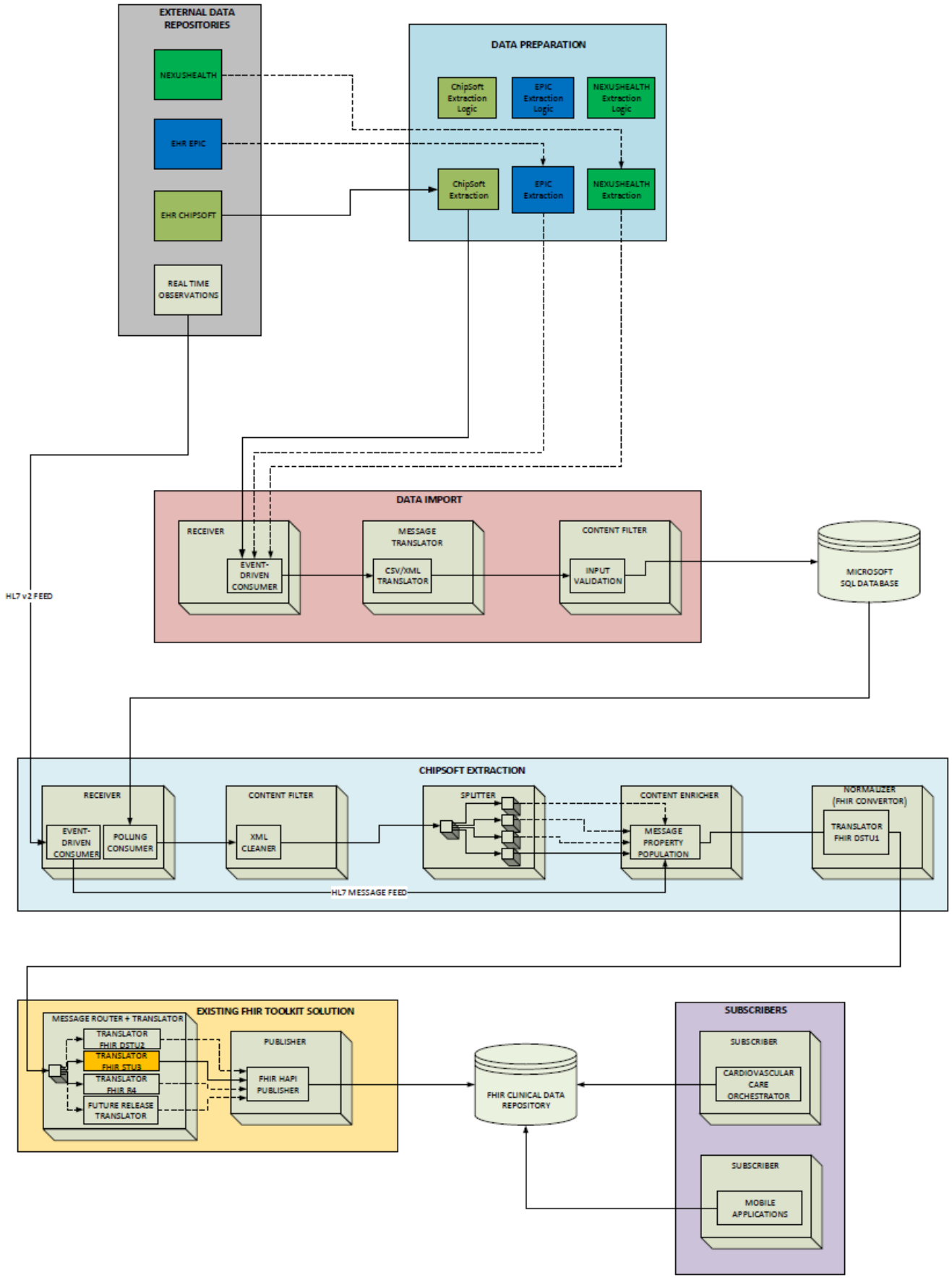

Figure 26: Solution Software Architecture

ness. Next, we want to investigate how to measure the prototype's efficiency based on the practical experiences obtained during the prototype's implementation phase [SRQ5a]. Ultimately, we want to estimate the impact of implementing the prototype in another healthcare institution and the effort required to scale up the prototype to process additional clinical concepts [SRQ5b]. We conclude this section by interpreting the obtained validation results.

8.1. FHIR STRUCTURE DEFINITION AND CONTENT VALIDATION

Recent advances in cardiovascular software solutions are looking for a FHIR-compliant data layer solution to build REST-full software architectures [Gøeg et al., 2018]. For this reason, we must ensure that after transformation, the most relevant FHIR resources are well structured according to the previously chosen data model. Additionally, we want to control that only specific FHIR resources may contain explicitly defined value sets of coded clinical concepts embodied in our FHIR data model. A validation process must ensure the correctness of the clinical data previously published by the software solution to the FHIR-CDR. By restricting the FHIR-CDR only to accept well-structured FHIR-compliant and appropriate data, we can present an accurate and unambiguously defined FHIR data layer to third-party software solutions [Mandel et al., 2016]. In this section, we explain how we applied the validation method to validate the structure and content of the generated FHIR output.

VALIDATION PROCESS

Initially, the intention was to use a docker image to host an FHIR HAPI CDR environment inside the prototype. We encountered some limitations during prototype validation because we could not adjust the docker image's validation level of the HAPI environment. To continue the research and better control the validation level, we decided to use the Vonk server, which is Firely's FHIR solution[36]. This FHIR server is an out-of-the-box solution, available on Simplifier.net[37], empowering us to control several features, including the validation level. Simplifier.net is an FHIR development collaboration platform, publishing software, tooling, and facilitating FHIR profile sharing, promoting efficient data exchange in healthcare.

Since FHIR offers the flexibility to model a particular clinical concept in many ways, we have set up a minimalistic FHIR compliant data model for this research to represent the extracted information, depicted in figure 19. The validation process needs to check various aspects defining explicitly the resource representing the extracted clinical concept. Important aspects describe the FHIR resource's hierarchy, considering the general composition of the clinical concept, and verify all data elements' conformance against the defined data types of the standard. Validation also verifies whether the allowed cardinality of the individual data elements meets the expected multiplicity. FHIR profiles define such a set of rules restricting an FHIR resource as specified by the FHIR specification. A FHIR resource subsequently declares conformance to a profile in its metadata profile element. Because this research aims to study how to design a scalable multi-institutional data extraction solution,

[36] https://fire.ly/
[37] https://simplifier.net/

we are not primarily concerned with defining the FHIR data model. This section investigates how to validate the FHIR resource's correctness created by our solution to comply with the FHIR standard specification. To validate the produced FHIR resources' hierarchy, we developed several FHIR profiles specific to our proposed prototype. The FHIR specifications published on the Internet defines a core platform suitable in a variety of clinical contexts. Based on these core FHIR specifications, we modeled the extracted EHR data concepts by constraining the specification according to the corresponding clinical usage context. Practically, the FHIR standard provides diverse conformance resources constraining the FHIR resource structure derived from the FHIR specifications. FHIR conformance resources reflect the adaptations applied to the FHIR core specification. A structure definition, named FHIR profile, is such an FHIR conformance resource that permits us to describe the structure, cardinality, and used data types of constrained FHIR resources applied in our solution. We published all generated and applied FHIR profiles to validate our FHIR resources on Simplifier.net under the project 'my graduation'.

Figure 27 schematically illustrates the validation process to verify our own developed FHIR resources' structure against the FHIR specification. The validation process starts by supplying the FHIR CDR with the FHIR conformance resources, highlighted in red, defining the expected FHIR resources' structure. After providing the FHIR Structure Definitions to the FHIR-CDR, we publish the prototype's FHIR output towards the FHIR-CDR. The outcome of this validation process returns the validation operation outcome reflecting the acceptance of the data instance into the FHIR-CDR.

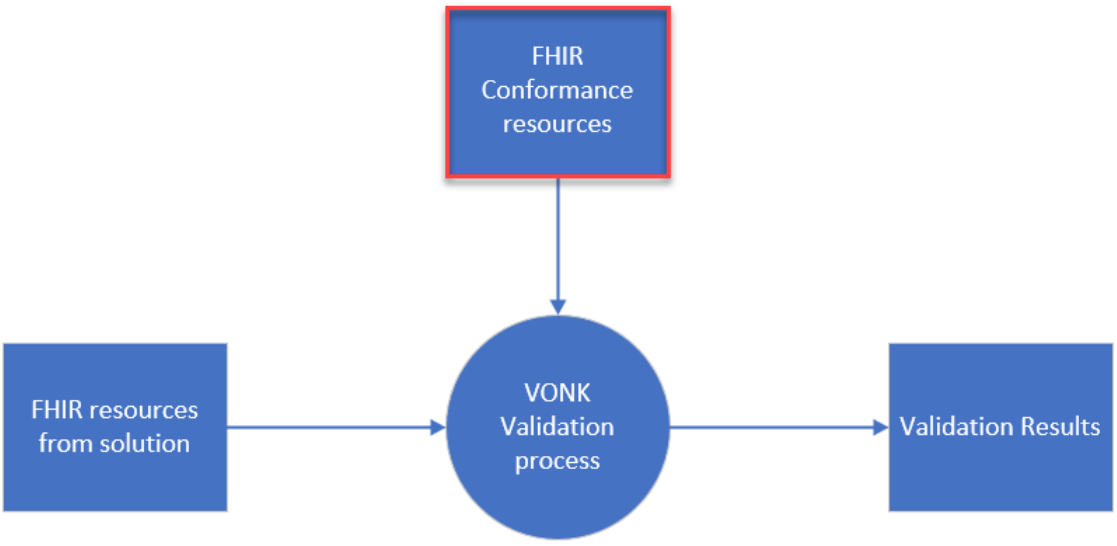

Figure 27: Validation Process

Implementation of the FHIR Validation Process

As mentioned earlier, Vonk enables us to manage the validation process. A configuration file empowers us to choose between three validation levels. Figure 28 depicts in red the three validation levels available in Vonk. By selecting the first validation level (OFF), we bypass the entire validation process. The FHIR-CDR does not perform any validation, and

the CDR accepts any FHIR resource without regard to the loaded FHIR profiles. A second option (CORE) allows us to increase the validation level. In this case, the only requirement is that the FHIR resources must conform to the FHIR core specification as published on the FHIR website [38]. By selecting the third option (FULL), the CDR performs validation against the core profiles and supplementary against the profiles defined within the metadata element of the FHIR profile. We applied the most restrictive option to validate the resources against our own developed FHIR profiles.

```
"Validation": {
  "Parsing": "Permissive", // Permissive / Strict
  "Level": "Full", // Off / Core / Full
  "AllowedProfiles":
[
"http://mygraduation.org/fhir/StructureDefinition/MyEncounter",
"http://mygraduation.org/fhir/StructureDefinition/MyOrganization",
"http://mygraduation.org/fhir/StructureDefinition/MyPatient",
"http://mygraduation.org/fhir/StructureDefinition/MyPractitioner",
"http://mygraduation.org/fhir/StructureDefinition/MyProcedure",
"http://mygraduation.org/fhir/StructureDefinition/MyObservationValueCodeableConcept",
"http://mygraduation.org/fhir/StructureDefinition/MyObservationValueDateTime",
"http://mygraduation.org/fhir/StructureDefinition/ObservationCreatinine_6050",
"http://mygraduation.org/fhir/StructureDefinition/ObservationCreatinine_8510",
"http://mygraduation.org/fhir/StructureDefinition/ObservationHD_Lipoprotein_6105",
"http://mygraduation.org/fhir/StructureDefinition/ObservationHeight_6000",
"http://mygraduation.org/fhir/StructureDefinition/ObservationHemo_CS00013233",
"http://mygraduation.org/fhir/StructureDefinition/ObservationHemoglobin_6030",
"http://mygraduation.org/fhir/StructureDefinition/ObservationHemoglobin_8505",
"http://mygraduation.org/fhir/StructureDefinition/ObservationTotalCholesterol_6100",
"http://mygraduation.org/fhir/StructureDefinition/ObservationTroponinT_6095",
"http://mygraduation.org/fhir/StructureDefinition/ObservationTroponinT_8520",
"http://mygraduation.org/fhir/StructureDefinition/ObservationWeight_6005",
...
]
}
```

Figure 28: Control Validation Level

To determine whether a specific FHIR resource, previously generated by the prototype, is compliant with a particular assigned profile, the FHIR standard included various validation functionality. Figure 29 presents a small-scale hierarchical overview of the involved FHIR resources implemented during the validation process of the FHIR Observation resource types employed by our prototype. We want to outline the validation process through two examples. In the first example (outlined in red), we briefly explain the validation process for validating an FHIR Observation of the type CodeableConcept. As discussed earlier, these are observations where we aim to validate particular clinical concepts defined in the ChipSoft standard content. As a second example (outlined in yellow), we describe the validation of an FHIR Observation classified as type ValueQuantity. This validation process mainly concerns the correctness of the measured value represented in an FHIR Observation resource. Before we can validate the generated FHIR CodeableConcept observation resources, we need to build additional FHIR resources to support the validation process and then load them into the FHIR-CDR of our software solution. The red-colored entities in figure 29 refer to applied FHIR ValueSet resources containing a list of authorized code sets defined within specific terminology standards. These ValueSet resources empower us to reject any FHIR Observation resource holding unspecified clinical codes. By applying this

[38] http://hl7.org/fhir/STU3/

method, we can guarantee that our software solution rejects unwanted clinical concepts. Next, we developed for each FHIR Observation of the type CodeableConcept a FHIR Structure Definition resource. The green depicted Structure Definition resources describe the underlying structure that the FHIR resources must conform to following the earlier agreed data model before the software solution finally accepts the observation resource. We applied a similar validation method in the second example. However, since ValueQuantity FHIR Observation types aim to contain quantified amounts expressed in arbitrary units, we will not be able to apply a FHIR ValueSet to validate the clinical content. Instead of defining a FHIR ValueSet resource, we defined a datatype profile where we can outline the required coding system and unit representation of the measurement. We depicted in figure 29 these datatype profiles in orange.

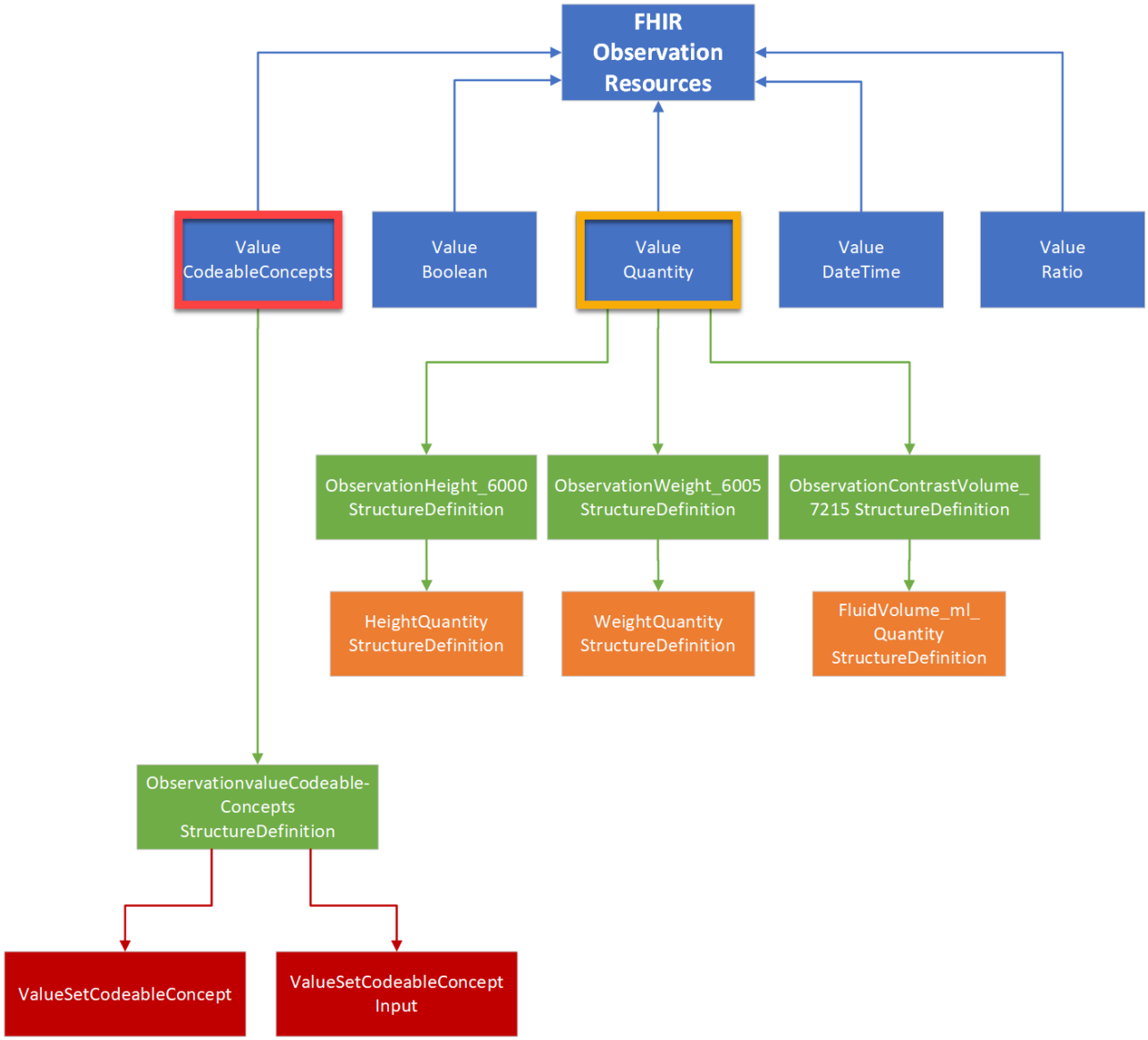

Figure 29: Observation validation

To elaborate on a clear presentation of a ValueQuantity FHIR Observation, we need to verify at least three different parameters to facilitate the unambiguous electronic communication and interpretation of clinical concepts.

1. Since our solution relies on clinical information captured and stored by other data sources, we want to ensure that the measured value or amount generated by our software solution is a realistic representation of the extracted value.

2. To ensure that the communicating parties attach the same meaning to the information exchanged, we must assign concise semantics to each defined unit. For this reason, we want to ensure that the base unit of the measurement corresponds to the base unit expected for this particular observation.

3. Since the language of medicine is complex, we want to ensure that our solution applies the correct terminology to represent the generated clinical concepts. Because we want to avoid errors introduced by manipulating lookup tables, we want to ensure that our solution only accepts terminologies intended for use in a particular context.

To verify these criteria in practice, we linked each FHIR Observation object of the type ValueQuantity with a single FHIR Structure Definition (green-colored), uniquely identified by name and observation ID (e.g., ObservationHeight_6000 StructureDefinition). This FHIR profile enforces the FHIR Observation structure, imposing restrictions on the corresponding clinical concept's coding and coding system. By restricting this structure, we can enforce that the patient's height is only accepted when the code and the associated coding system reference the correct values defined by the terminology system. This way, we can guarantee that a patient's height always points to the correct code within a predefined terminology system. Next, the observation height Structure Definition refers internally to another datatype profile, depicted in orange, restricting the default FHIR quantity value type. By enforcing a restriction on the default FHIR quantity type, we can explicitly restrict specific conditions expressing the patient's height. Figure 30 represents an example where we profile the Quantity data type by restricting some elements. We indicate that the unit element, representing a human-readable unit, must be expressed in centimeters. Also, the element system must contain a reference to the Unified Code for Units of Measure. We restrict this by enforcing the element to use a URI pointing to the corresponding code system (http://unitsofmeasure.org). Finally, we indicate in the code element how we want to express the computer processable form of this quantity datatype. Additionally, the quantity type profile allows us to configure a range defining a minimum and a maximum expected value of the patients' height. Figure 46 of the appendix includes an XML representation of a structure definition illustrating the restrictions mentioned above and highlights all modifications in yellow. The FHIR-CDR rejects all FHIR resources that do not comply with the imposed restrictions.

The software solution offers a simple solution to declare a FHIR resource conformance through a particular profile by populating each FHIR resource's metadata element with a profile URL. Figure 31 demonstrates how we designed the solution to assign an FHIR profile to a particular FHIR resource through a lookup table. The canonical name defined in the lookup table determines the corresponding FHIR resource's compliance with a specific

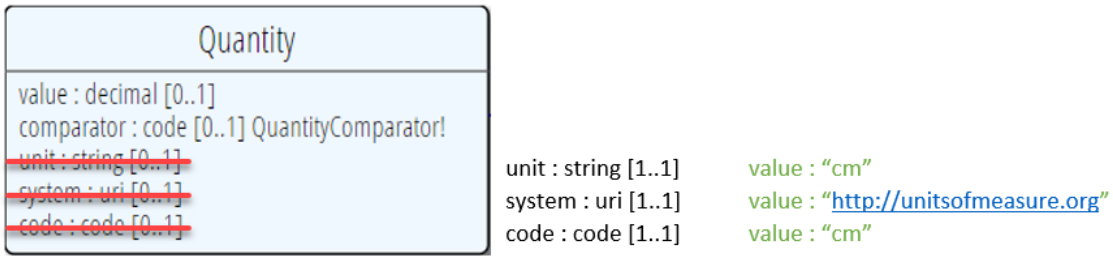

Figure 30: Restriction of Quantity datatype

structure definition. This design empowers us to adapt the validation process quickly without making changes to the code.

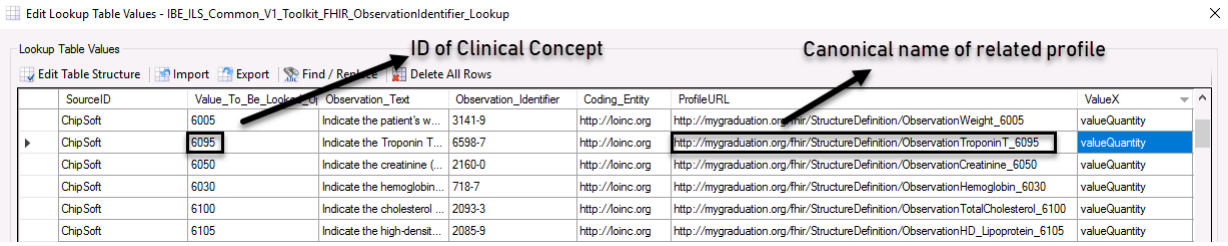

Figure 31: Solution to link a ProfileURL to a specific clinical concept

To avoid errors when constructing FHIR profiles manually due to the high error proneness, we decided to use the Forge[39] FHIR profile editor tool distributed at Simplifier.net to create and edit FHIR profiles graphically. Forge easily allows us to browse through the FHIR element tree structure graphically and automatically creates JSON structure definition resources to impose restrictions on the FHIR specifications. Since the Forge tool integrates with Simplifier, we have published the FHIR profiles against our public project. Publishing the profiles empowers us to validate the Structure Definitions against the FHIR resources generated by our software solution. The ability to manually validate the created FHIR resources before publishing them to our test CDR permits us to build the Structure Definitions gradually. Invoking the online Simplifier validation tool[40] helps us identify problems during the Structure Definitions' creation process and allows us to validate the conformance of the generated resources against the specifications defined by our profiles. Once the profiles and valuesets are in place and tested using the online validation tool, we can preload our Vonk test CDR environment with the customized conformance FHIR resources to constrain the default FHIR specifications. We can control all conformance resources through an administrative FHIR API to manage our prototype's validation process. For this study, we used Postman[41] to load the conformance resources to the Vonk test FHIR server. Postman is an HTTP post client tool that we can use to perform basic create, read, update and delete (CRUD) operations on the FHIR CDR. Figure 32 represents an example of such an operation to send a conformance FHIR resource to the CDR through a POST HTTP operation.

[39] https://fire.ly/products/forge/
[40] https://simplifier.net/validate
[41] https://www.postman.com/

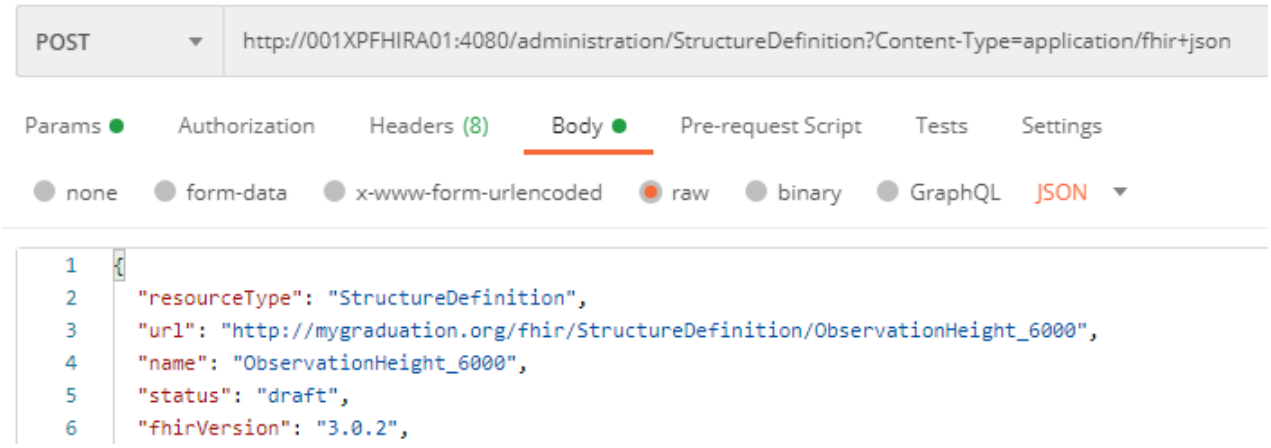

Figure 32: Example of Postman POST operation to send a structure definition to test CDR.

The procedure described above allowed us to validate all published FHIR resources for syntactic and semantic correctness. Based on the applied validation method, we have detected several errors during implementation related to syntax violations. After making the necessary adjustments in our prototype, we could successfully validate all published FHIR resources in the FHIR CDR.

8.2. Scalability validation

Since our study results indicate that there is still a perspective to expand the cardiovascular dataset, leading healthcare organizations such as Philips desire to estimate in advance how much time it will take to extend the prototype to a broader dataset. This subsection briefly lists the necessary activities required to prepare our software solution to process new cardiovascular concepts. Finally, we also explain how we can verify the prototype's scalability among other clinical environments implementing the ChipSoft EHR.

In case the ChipSoft data extraction tool offers a new CSV file that includes additional clinical observations, we need to make some adjustments before the prototype can process the newly added concepts. A fundamental change in the data import route is required to load the CSV file's new composition, including recently added content, into the SQL database. This change mainly involves adapting some SQL statements to store the CSV content in the database correctly. Once the additional information is available in the database, the polling mechanism, located in the ChipSoft extraction route, requires a minor modification through altering the query to incorporate the new content in the XML structure. As a result of the extended XML structure, we have to adapt the XML schema of the new incoming data structure to offer the normalized data transformer subsequently with this schema. After implementing these minor changes, the software solution can process the newly added cardiovascular concepts automatically, given that specific lookup tables require alterations to process particular clinical concepts. These adjustments depend on the FHIR Observation type. It is important to note that we focus on automated processing of clinical concepts classified as FHIR Observations to reduce implementation time. Clini-

cal concepts not classified as FHIR Observations always require manual intervention in the code. Despite the applied modularization technique, it is possible to locate precisely where code adjustments are needed. A correct classification must determine the observation type of each particular clinical concept. Figure 29 represents some instances of possible observation types in blue (e.g., valueBoolean, valueQuantity, valueCodeableConcepts,...).

Based on the just discussed efforts needed to go through the entire ETL process, we have developed a cost model to determine the software solution's efficiency [Naik and Nayak, 2017]. This model can symbolize the total costs we have to consider while implementing the current prototype in another healthcare environment using the ChipSoft EHR. Additionally, the cost model can also provide a price indication if we have to accommodate more clinical concepts than currently covered by the prototype. We developed the cost model based on the practical knowledge gained during the prototype development process. These insights have given us an indication of the time required to develop the prototype. To further refine the cost model, we listed the expected fixed cost based on the time necessary to shape an individual clinical concept towards its corresponding FHIR resource. To achieve this, we measured a unit price for each activity during development activities. The unit price represents the amount of time required, expressed in hours, to fulfill one single data modeling activity. After classifying the clinical concepts presented earlier in figure 17 and figure 18, we gained a better overview of the distribution and the number of FHIR resources required to model our obtained dataset from the ChipSoft framework. To estimate a representative calculation of the time necessary to transform additional clinical concepts by the prototype, we referenced our time calculations on the FHIR resource allocation ratio into the prototype's covered dataset. Using this ratio, we calculated this FHIR resource's weight in the enlarged portion of the dataset. To determine the FHIR resource transformation cost in the enlarged portion, we need to multiply the weight by the previously allocated action's unit price. Finally, by accumulating all the costs needed per activity, we obtain the project's total cost in hours. In figure 33, we applied a cost calculation method to determine the total project cost for three use cases. The first calculated cost (yellow-colored column) represents a baseline cost representing the development effort, expressed in hours, needed to build our prototype. A second reference price indication shows the cost to implement the prototype in another hospital using the ChipSoft EHR (green-colored column). The latter cost indication estimates the time required to expand the prototype processing fifty additional clinical concepts (orange-colored column).

Based on this cost model, we can conclude that the significant investment costs can reimburse themselves by implementing the prototype in other healthcare facilities using the same EHR. The implementation cost is principally limited to setting up an integration server hosting the software solution, loading the software configuration, and performing some basic configuration activities. The standardized representation of clinical concepts stored inside an EHR platform strengthens interoperability, empowering us to align our prototype in advance to the supplied dataset by the EHR. A similar study should further investigate whether a similar extraction approach applies to other EHR vendors. Based on the cost model analysis of figure 34, expanding the existing prototype with fifty additional cardiovascular concepts, we can deduce that a considerable time is required to develop the ChipSoft query (blue-colored - 37,5 hours). This time is mainly spent collecting the stan-

Figure 33: Cost Model

Cost Model

			Price / unit	BASELINE	REFERENCE DATASET	EXTENSION 50 DATACONCEPTS
Required activities	**Subactivities**	**FHIR resource types**		**Time spent to build baseline (hours)**	**New installation (hours)**	**Extension existing system (hours)**
Server setup (Fix cost)			*10*	10,00	10,00	0,00
Load configuration integration solution (Fix cost)				1,00	0,50	0,00
Development ChipSoft query			*0,75*	50,25	0,00	37,50
Load configuration ChipSoft extraction tool (Fix cost)			*0,5*	0,50	0,50	0,50
Development solution (Fix cost)			*250*	250,00	0,00	0,00
Initial configuration solution (Fix cost)			*1*	0,00	1,00	0,00
FHIR Resource identification			*0,1*	6,70	0,00	5,00
Adapt configuration solution						
	Required changes to get CSV data in SQL database		*0,02*	0,00	0,00	1,00
	Create new XML schema (Fix cost)		*0,5*	0,00	0,00	0,50
	Changes required in lookup tables	Patient	*0,2*	0,00	0,00	0,90
		Encounter	*0,2*	0,00	0,00	0,60
		valueQuantity	*0,2*	0,00	0,00	1,64
		codeableConcept	*0,25*	0,00	0,00	1,87
		valueBoolean	*0,2*	0,00	0,00	2,84
		valueDateTime	*0,2*	0,00	0,00	1,04
		valueRatio	*0,2*	0,00	0,00	0,75
		FamilyMemberHistory	*0,2*	0,00	0,00	0,15
		Procedure	*0,2*	0,00	0,00	0,60
	Required code changes	valueQuantity	*0,75*	0,00	0,00	0,56
		codeableConcept	*0,75*	0,00	0,00	1,68
		valueBoolean	*0,75*	0,00	0,00	0,00
		valueDateTime	*0,75*	0,00	0,00	0,56
		valueRatio	*0,75*	0,00	0,00	0,00
Mapping FHIR Resources (excl. Observations)	Patient		*1*	0,00	0,00	4,48
	Encounter		*1*	0,00	0,00	2,99
	FamilyMemberHistory		*1*	0,00	0,00	0,75
	Procedure		*1*	0,00	0,00	2,99
Link local concepts to terminology standards		valueQuantity	*0,4*	0,00	0,00	3,28
		codeableConcept	*1*	0,00	0,00	7,46
		valueBoolean	*0,4*	0,00	0,00	5,67
		valueDateTime	*0,4*	0,00	0,00	2,09
		valueRatio	*0,4*	0,00	0,00	1,49
		Total		318,45	12,00	88,87

dardized clinical concepts within the EHR platform and subsequently representing them in the desired fashion through the query. A significant amount of knowledge of the EHR structure and clinical experience is required to complete the extraction. Once the baseline query is available, we can reuse the query in other ChipSoft EHR environments and develop it further incrementally. A second significant cost is mapping clinical concepts to international coding standards (dark blue-colored - 20 hours). The locally defined clinical concepts included in ChipSofts' standard content dataset require a great deal of effort to align them unambiguously with international coding standards. A possible proposal to suppress these costs is to use terminology solutions automating the mapping process. State-of-the-art technology can support the mapping process by reducing the time needed to accomplish mappings between different coding schemes. Additionally, terminology solutions have the advantage that the previously locally managed translation tables can be preloaded centrally, so centralized management can keep these terminology solutions maintainable and up-to-date with the most recently published codes [Metke-Jimenez et al., 2018].

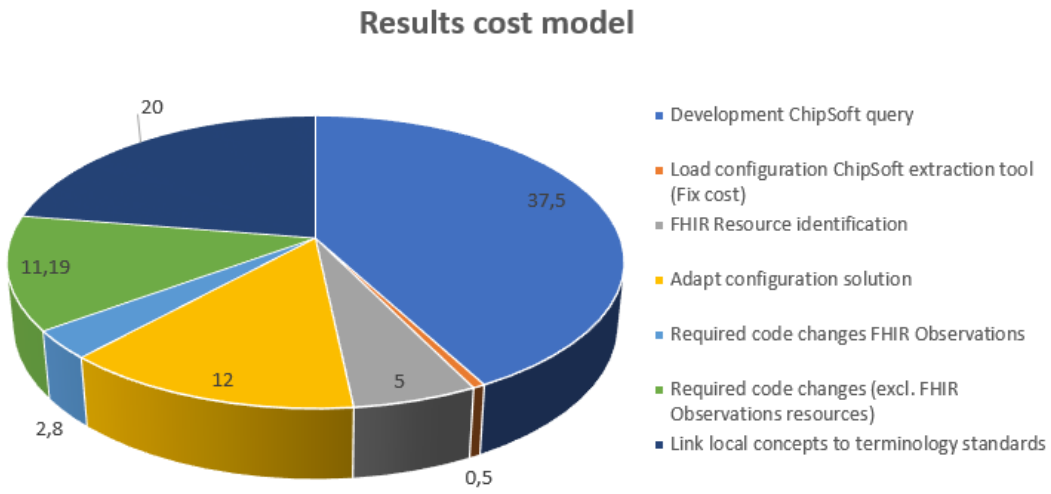

Figure 34: Results Cost Model

A third challenge (orange-colored - 12 hours) denotes the time required to adapt the solution to process new data. Connecting the prototype to a new dataset always requires an adjustment to adapt the solution's configuration. Some changes are necessary to get the CSV information into the SQL database and align the new XML schema with the new dataset. Moreover, we recognize that manual interventions are still essential by configuring lookup tables to shape the extracted information according to the FHIR specification. However, we also recognize that specific clinical concepts need code modifications to transform them into FHIR resources. We note minor code modifications needed for the FHIR Observations resources tackled within this study (light blue-colored - 2,8 hours). In contrast, we recognize that other FHIR resources demand more code modifications to model the clinical information according to the FHIR specification (green colored - 11,19 hours).

If we want to estimate our prototype's efficiency, we can emphasize the clinical concepts modeled as FHIR Observation resources. These represent the vast majority of the

clinical concepts in the initial dataset. By focusing on these FHIR Observations, we can determine the weight of the number of resources *not* requiring any code modification. Figure 35 shows the FHIR Observations weight distribution where we need to adapt code to realize the transformation to the FHIR specification correctly, colored in orange. Based on these numbers, we can apply the formula depicted in figure 36 below, outlining the weighted arithmetic mean representing the cost resulting from the time saved avoiding a code change.

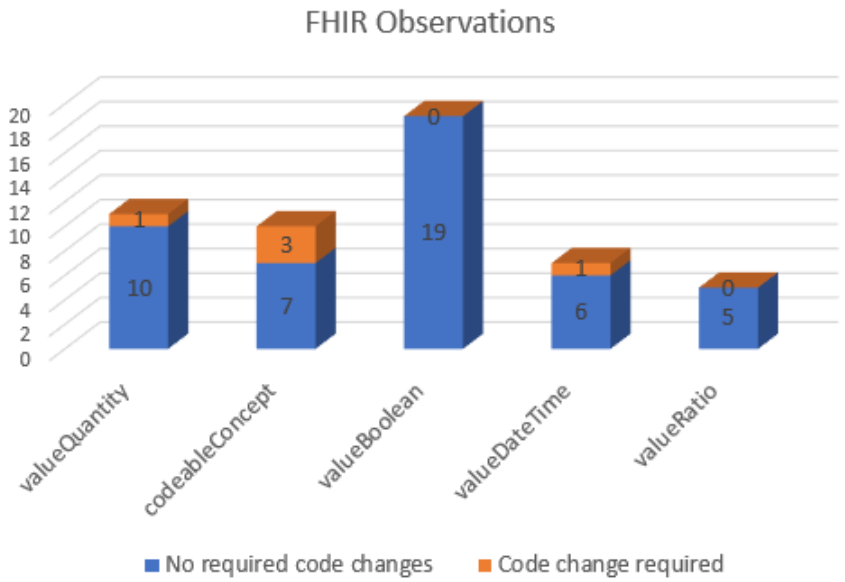

Figure 35: Weight Distribution FHIR Observation resources related to required code changes

$$\overline{x} = \frac{\Sigma_{i=1}^{n} g_i . x_i}{\Sigma_{i-1}^{n} . g_i}$$

Figure 36: Applied formula to calculate weighted arithmetic mean

We can classify, in total, from the 67 clinical concepts extracted by the ChipSoft query, 52 clinical concepts as FHIR Observations. Consequently, this means that we can process 47 of the 52 observations in total without making code changes. This amount represents a significant amount compared to other FHIR resources where each resource requires a code modification. Suppose we calculate the weighted arithmetic mean of all FHIR Observations that do not require code adjustments related to the total number of modeled FHIR resources. In that case, we can conclude that approximately 37 FHIR Observation resources are eligible if we round up the results. In figure 37, we illustrate the formula's application to calculate the arithmetic mean to our findings within this research context.

Suppose we assume that the fixed unit price of an FHIR Observation classified clinical concept is 0.75 hours. In that case, we can determine that the prototypes' efficiency is approximately 28 hours (37 FHIR Observation resources * 0.75 hours), indicating we can save

$$x = \frac{0.6 + 0.4 + 47.52 + 0.1 + 0.4}{6 + 4 + 52 + 1 + 4} = 36.48$$

Figure 37: Calculate weighted arithmetic mean

about 28 working hours by grouping all FHIR resources and adapting the prototype transformation process accordingly.

To gain insight into the prototype's scalability among other clinical environments implementing the ChipSoft EHR, we examined the results after transferring the standardized query between different medical settings. We involved another Belgian medical institution in loading our developed query into their ChipSoft EHR software platform. Consequently, we investigated the query's results by examining whether the prototype can interpret the returned data set. A practical test illustrates that the prototype can perfectly integrate the returned dataset. Next, we analyzed the differences between the ZOL Genk dataset and the other medical institution dataset. This analysis reveals that not all hospitals record the same amount of data in their EHR. We concluded that the other Belgian healthcare institution registered 37 fewer clinical concepts for the same query compared to ZOL Genk. After investigation, we revealed the hospital is currently transitioning to register all data centrally within the EHR. Within the framework of government guidelines regarding quality registration and reimbursed treatments, various customized interfaces are operational to deliver required information from a local cardiovascular information system to quality organizations. Furthermore, government policies require clinical information delivery for quality purposes through a dedicated healthcare platform [Delvaux et al., 2018]. Currently, the lack of an automatic bi-directional delivery mechanism hinders the interfacing with EHR software platforms.
In addition to the ongoing transition phase, we also note that not all clinical departments integrate the same information within the EHR. Differences in clinical workflows among hospitals can affect the EHR integration level justifying the data variation. For example, we refer to the EHR integration of the hemodynamic (blood flow analysis) data within ZOL Genk. During this project, a separate integration project was ongoing to integrate this information into the EHR, illustrating that the lack of EHR integrations across different hospitals causes an unbalanced amount of cardiovascular data in different EHR platforms.

8.3. Performance validation

Since diverse software applications can take advantage of up-to-date unambiguously defined medical data, we want to overview the number of messages the prototype can handle. For example, before a clinical dashboard can refresh some data, the information provided must go through the entire ETL process before the prototype can make the data unambiguously available for third-party software applications. For this reason, the speed to complete the ETL process and the update intervals in which we supply new data are essential factors of the prototype. To determine the processing speed at which the prototype can extract data and transform it into unambiguous information, we subjected the prototype to a performance test. Initially, we systematically duplicated the incoming data set, consisting of

some cardiovascular interventional report extractions, by simulating additional PHI data using a patient generator[42]. A significant advantage of using such a tool is that we can generate patient data without worrying about legal and privacy restrictions. The patient generator's tool disadvantage is that we cannot produce clinical information comparable to the ChipSoft extraction tool's data. This restriction prevents us from reproducing the existing variation in clinical data from real-world situations. Since we aim to simulate the existing variety of registered clinical concepts to perform representative performance tests, we completed the test with real data from cardiovascular interventional studies. To ensure the patients' legal and privacy restrictions, we performed performance tests only on the hospital's test environment.

We initiated the performance tests based on a data set representing approximately 550 cardiovascular studies. Next, we repeatedly submitted this complete dataset to the prototype for further processing over eight hours. Subsequently, we analyzed the message queue within the integration platform's monitoring tool. By adjusting the repeated pattern frequency, which determines how often we deliver a test dataset to the software solution, we could visually determine when message queuing occurs in the prototype. The moment messages queuing occurs represents the prototype's pivot point, indicating that the message throughput has reached its limit and further message processing starts slowing down. Figure 38 depicts the frequency pattern indicating the timestamps sending in batches containing 550 cardiovascular reports extractions from the ChipSoft EHR.

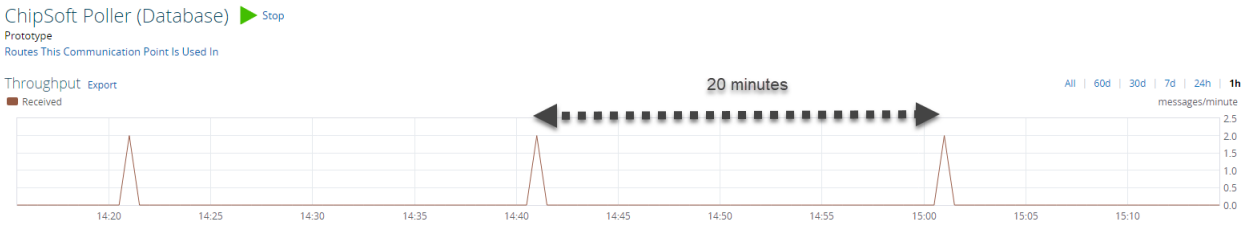

Figure 38: Frequency pattern for submitting test data in batches.

Performance tests show that the prototype can process every 20 minutes a batch of 550 cardiovascular exams without message queuing. Figure 39 shows the graphical representation of the throughput measured at the outbound communication point in response to the supplied batched dataset at a time interval of 20 minutes. The visualization shows that a new batch is only delivered after the prototype has finished processing the previous dataset. In the short zones where we do not register data throughput, we provide in real-world situations the necessary time to convert new (non-existing) CSV content to the SQL environment.

From this result, we can conclude that we need approximately two seconds to complete the entire ETL process holding 67 clinical concepts. It is important to mention that most of the time is required to publish the data to the FHIR CDR (load phase of the ETL process). Since we use an HTTP client CP to establish a request to the FHIR-CDR using the HTTP protocol, we apply synchronous communication to publish the messages. The use

[42]https://syntheticahealth.github.io/synthea/

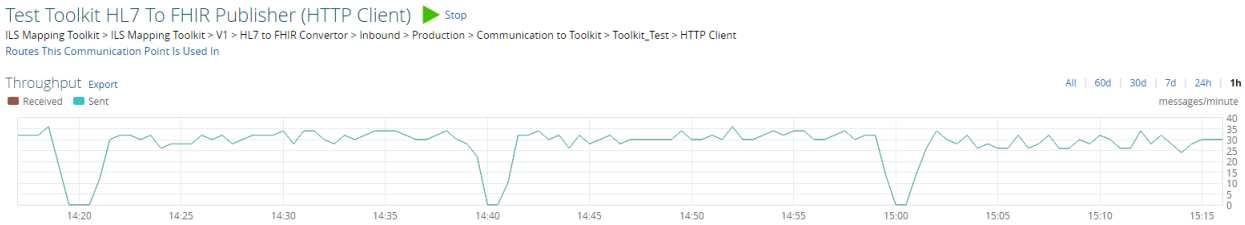

Figure 39: Throughput data publisher in response to 20 minutes input frequency.

of synchronous communication means that our solution publishes a FHIR message to the FHIR-CDR immediately after the CP receives a message from the integration engine. After getting back a successful processing code from the FHIR server, the HTTP client can deliver the subsequent FHIR message produced by the prototype.

As earlier stated, we have indicated that we want to consider possible report adjustments performed by cardiologists later than the study date. For this reason, we proposed to schedule a daily batch containing the cardiovascular reports of the last seven days. This proposal suggests that our prototype could handle a daily maximum of roughly 78 (550 cardiovascular reports / 7 days) cardiovascular studies within 20 minutes. Estimating an average PCI procedure time of 40 minutes Mahmud et al. [2017], we can assume our prototype can manage this PCI procedure rate. Considering that we performed the tests on a test environment with evaluation software where we could not increase the number of simultaneous connections to the FHIR-CDR, we can assume that the processing speed is much higher in practice. Additionally, the processing of this batch of PCI-related information requires only a short-term processing capacity of the underlying IT environment. We can fill the idle time of the prototype by processing in parallel HL7 v2 messages. Since we expect that this mainly concerns measurement results, broadcasted at any time during procedure time, we do not expect any performance problems.

9. Conclusions

This study investigated how to develop a scalable software architecture to collect cardiovascular information from multi-source clinical repositories with minimal integration effort. The goal was to develop a reliable solution to unambiguously represent a subset of cardiovascular information, specified in the NCDR CathPCI Registry dataset, according to international terminology standards to boost data interoperability and clinical software development. The ability to deploy the solution in diverse medical institutions aimed to reduce healthcare organizations' high implementation costs and ultimately make integration projects more profitable.

Our main research question concerned the investigation of designing a scalable and multi-institutional deployable data integration solution for the collection and transformation of cardiovascular data towards an unambiguously defined dataset. We have presented an answer to this main research question by clarifying the problem definition through answering five sub-research questions.

The first research question deals with investigating in which clinical repositories we can access cardiovascular information. We focused on a reference dataset listing several cardio-

vascular concepts. Investigation revealed that healthcare providers accommodate an enormous amount of clinical data captured from diverse siloed data repositories. The medical institution's data-driven mindset participating in this study shows a high degree of cardiovascular data integration with the hospital-wide information system (EHR). However, we cannot assume that all medical institutions integrate cardiovascular information to the same extent into the EHR. Covid-19 has prevented us from obtaining a detailed view of various multi-vendor clinical data repositories that may contain cardiovascular information in a multi-institutional environment. On the other hand, we recognize that many clinical data repositories still exchange information through HL7 v2 communication but lacking data integration with hospital-wide information systems. Because this study aimed to investigate a multi-institutional deployable solution and hospitals facilitate an integrated EHR system for the central storage of clinical information, we focused on these EHR software platforms to extract cardiovascular data. To outline the multi-vendor EHR landscape across Belgian (Flanders) and Dutch healthcare providers, we conducted a market analysis and found several existing EHR vendors in both markets. Despite the great variety, we can conclude that ChipSoft is the only prominent EHR vendor active in both markets. Interestingly, ChipSoft is only active in the Belgian and Dutch markets, while Epic is active globally. Due to time constraints and the ongoing Covid-19 pandemic, we have focused exclusively on the ChipSoft EHR as a clinical data repository to extract cardiovascular information. Although we focus on hospital-wide information systems to extract and transform cardiovascular information in this study, we still provide the ability to collect and transform cardiovascular data from various clinical data sources based on HL7 v2 data streams delivered by those clinical data repositories.

For the second sub-research question, we explored an appropriate extraction method to extract cardiovascular information subject to variation in diverse clinical settings. Together with different EHR vendors and the collaborating hospital where we implemented a prototype, we have investigated various extraction methods. We concluded that standardization is a fundamental concept to realize a data extraction within diverse environments subject to change. Since hospitals do not always have the appropriate reporting and data analysis objectives and do not comply with standardized extraction methods, we opted for a vendor-specific extraction technique to extract the EHR data in a standardized way. We have discovered that the ChipSoft EHR provides a data extraction tool able to create data overviews immediately derived from the EHR data platform. Furthermore, this extraction method offers the opportunity to export and import developed data extraction overviews between different clinical environments. The result of this extraction method shapes the input dataset aligned with the developed multi-institutional deployable prototype.

The third research question deals with the representation of cardiovascular information across different clinical information systems and clarifies which fraction of cardiovascular data is in a structured format. An analysis of the data shows that ChipSoft uses vendor-specific standard content to express clinical concepts. The standard content is a vendor-specific coding system that introduces uniformity to approach diverse clinical concepts in different medical settings and is an essential concept within this research. Our study results illustrate that by fully aligning our extraction query on standard content, we can extract identical clinical concepts from different clinical environments in a standardized

way to deliver a uniform dataset to the developed software prototype. Out of 345 clinical concepts defined in the NCDR cathPCI data dictionary, we were able to find 67 standardized clinical concepts in the ChipSoft EHR, representing 20% of the information listed in the cathPCI data dictionary. However, we believe that the Covid-19 pandemic may have slightly affected this outcome since we could not rely on cardiologists' expertise to extract more complex clinical entities. Under normal conditions, we expect to be able to increase the amount of cardiovascular data to 35%.
Due to the ever-increasing collaboration between medical institutions and the drive for standardization in healthcare, strongly motivated by the Covid-19 pandemic, we expect to see an increasing trend in the number of cardiovascular concepts extracted from EHR systems in the coming years. Despite, we also observe large differences in the amount of registered data in the EHR among different healthcare organizations. Results show that for the same standardized query, 55% fewer results are returned in a different clinical setting. After discussing these results with this medical organization, the hospital IT management claims an ongoing transition towards a full EHR integration of all siloed cardiovascular data stores. The main reason is that dedicated government healthcare platforms do not yet support a bidirectional delivery mechanism for an automatic reporting interface between software applications and government entities. Since hospitals recognize the importance of a central registration platform, the lack of such an interface implies that hospitals do not yet require mandatory integration of all CDRs with the EHR.

The fourth sub-research question concerns how we can efficiently deal with data transformations to express cardiovascular information in an interpretable fashion. To answer this research question, we developed a software prototype to align our proposed software architecture to a realistic clinical work environment. We have found that the FHIR standard gains widespread support and momentum among healthcare developers due to the detailed FHIR specification guidelines and the lightweight atomic data access design. The developed prototype is fully aligned with the FHIR standard and offers a flexible way to generate unambiguously defined data sets for various (future-proof) FHIR standards. Additionally, we discovered that clinical terminologies are fundamental to transform clinical concepts into internationally recognized meaningful data concepts interpretable by humans and computers. In compliance with this existing FHIR standard, we developed the prototype to transform the extracted cardiovascular concepts into unambiguously defined datasets suitable for diverse objectives to improve data interoperability within healthcare. During the prototype development, we discovered some inefficient message handling techniques preventing efficient message transformation. We tackled these inefficiencies by developing a query that immediately represents the extracted clinical concepts in an organized fashion to improve data transformation efficiency. The development of the prototype has revealed some shortcomings in the management of clinical terminologies. Linking proprietary clinical concepts with internationally recognized terminologies is a time-consuming task, which also requires clinical knowledge. However, we can conclude that the prototype's applied design is not maintainable if used in various multi-vendor clinical environments. We recognize difficulties in maintaining the locally managed translation tables when scaling up the number of clinical concepts. Additionally, we envision a rising complexity for the management of semantic data interoperability due to an increased number of EHR software platforms supported by our prototype. Research results show that

it is better to deploy centrally managed terminology solutions to keep the solution maintainable and prevent possible translation errors due to human error by expanding semantic mappings. In addition to the realized local semantic mappings applicable for the ChipSoft EHR, enterprise-wide terminology solutions can contain semantic mappings based on international coding standards. Since these terminology solutions also offer the possibility to publish semantic mappings to local terminology solutions, this can introduce enormous advantages to keep locally deployed terminology solutions up-to-date with the most recent semantic mappings. In addition to the proposal to implement a centrally managed terminology solution, we have analyzed in this study how we can express clinical concepts spanning different vocabularies. We addressed this problem with UMLS and explored how this solution could fit into our prototype. Due to technical and security restrictions, we did not integrate UMLS in our prototype but only demonstrated how to obtain a uniform UMLS lookup.

The last sub-research question clarifies how to measure the proposed solution's efficiency to determine the prototype's impact on integration costs. As a first indicator, we checked the output of the software solution towards correctness. We went through a validation process to validate the generated FHIR resources on both structure and content. Before going through the validation process, we developed several FHIR profiles for each clinical concept extracted from the ChipSoft EHR. By linking the corresponding FHIR profiles to the generated FHIR resources, we successfully validated all FHIR resources created by the software prototype. By completing this validation process, we can guarantee that all output generated by the proposed software solution represents realistic and unambiguously defined information according to international terminology standards. A second essential indicator concerned validating the scalability of the software solution. We investigated the prototype's scalability by increasing the number of extracted cardiovascular concepts delivered to the prototype and analyzed the extent to which the multi-institutional solution returns cardiovascular data after deployment in another healthcare institution. To answer this question, we have built a cost model based on some software quality characteristics. This cost model estimates the hours required to implement the current prototype in a different clinical setting. Moreover, this cost model estimates the number of hours needed to expand the current prototype if more clinical concepts have to be processed. During the development process, we found that most of the time is required to collect clinical concepts from the EHR platform. We found that this requires a thorough knowledge of the EHR structure and an excellent background of the context of all clinical concepts. Additionally, we concluded that aligning all extracted clinical concepts with internationally recognized terminology systems requires an enormous amount of effort. However, we determined that this investment cost can reimburse itself because the prototype is relatively easy to implement in other clinical environments. The cost model results show that we need barely 12 hours to deploy the prototype in a different medical setting, including integration server setup. Further analysis shows that implementing the proposed software solution reduces implementation time by 28 hours through adjusting the prototypes' extraction method and optimizing the subsequent transformation process. When we expand the prototype with about 50 additional clinical concepts on top of the currently obtained data set, our cost model reports that we still need approximately 89 hours to implement this modification. Investigation reveals that we could predict a drastic decline in the prototype's implementa-

tion time by relying on a centrally managed terminology solution. Further research should provide a clear indication of the explicit time savings. Since this cost model concentrates on the extraction of cardiovascular data from the ChipSoft EHR, similar research should demonstrate whether similar conclusions apply to other EHR vendors. The last indicator used to determine the prototype solution's impact is the processing speed to go through the entire ETL process. To validate performance, we subjected the prototype to a stress test in which we presented a ChipSoft EHR extraction holding 550 cardiovascular studies, each consisting of 67 clinical concepts. We could determine our prototype's maximum throughput rate by adjusting the polling frequency defining the rate at which we extract data from the ChipSoft EHR. We visually determined that message queuing occurred after delivering a cardiovascular dataset every 20 minutes towards the prototype. Based on these results, we can conclude that the prototype takes around two seconds to go through the entire ETL process for one cardiovascular report extraction. However, we remark the bottleneck mainly occurs while delivering the output of the prototype towards the FHIR-CDR. We can explain this behavior since we practice with evaluation software for the simulation of the FHIR CDR. Despite the use of evaluation software, we can conclude that the prototype's processing speed is more than adequate to meet a production environment's requirement to extract cardiovascular data from the ChipSoft EHR.

10. Future Work

This research mainly focused on developing a prototype to extract cardiovascular data in a multi-institutional environment. We discovered that hospitals centrally record and collect their clinical data originating from various clinical data repositories. Market research revealed that different EHR vendors are active in Belgian and Dutch healthcare organizations. Although this research focuses on extracting cardiovascular information from the ChipSoft EHR, the aim is to develop a multi-vendor compatible prototype able to process cardiovascular information originating from multi-vendor EHR providers. As Epic is a global EHR provider, the intention is to expand the research into the American market, making the solution globally deployable. Another interesting possibility we want to explore is a further investigation of the terminology solution. Because it was not allowed to connect our prototype with an external terminology solution in the clinical test environment, and we had no budget for a local test deployment of a local terminology solution, no additional tests were possible. We limited the practical implementation to some minimalistic tests. Ideally, we want to implement the prototype in combination with a test terminology solution where it is possible to import the local terminology mappings within a local or remote terminology solution. This setup should give us a better view of the maintainability of the solution. Furthermore, additional research using state-of-the-art technologies, supported by terminology solutions, can provide a better insight into whether these contribute to the effort required for unambiguous mapping of clinical concepts to international terminology standards. Based on the cost model results, we see the potential to develop the prototype further to allocate extracted clinical concepts automatically. In our opinion, the automatic recognition of FHIR observations can significantly improve the prototype's efficiency. Ongoing research by Kiourtis et al. [2019] shows the economic impact of this technology. Future integration of this technology can give more insight into this research's feasibility in combination with our prototype.

BIBLIOGRAPHY

Hina Abrar, Syed Jawad Hussain, Junaid Chaudhry, Kashif Saleem, Mehmet A Orgun, Jalal Al-Muhtadi, and Craig Valli. Risk analysis of cloud sourcing in healthcare and public health industry. *IEEE Access*, 6:19140–19150, 2018. 48

Ebtsam Adel, Shaker El-Sappagh, Sherif Barakat, and Mohammed Elmogy. Distributed electronic health record based on semantic interoperability using fuzzy ontology: A survey. *International Journal of Computers and Applications*, 40(4):223–241, 2018. 6

Jozef Aerts. Towards a single data exchange standard for use in healthcare and in clinical research. In *eHealth*, pages 55–63, 2018. 41

Awais Ashfaq and Slawomir Nowaczyk. Machine learning in healthcare–a system's perspective. *arXiv preprint arXiv:1909.07370*, 2019. 4

Ana Avilés-González, Juan Piernas, and Pilar González-Férez. Batching operations to improve the performance of a distributed metadata service. *The Journal of Supercomputing*, 72(2):654–687, 2016. 45

George W Beeler. HL7 version 3—an object-oriented methodology for collaborative standards development. *International Journal of Medical Informatics*, 48(1-3):151–161, 1998. 7

Ashwin Belle, Raghuram Thiagarajan, SM Soroushmehr, Fatemeh Navidi, Daniel A Beard, and Kayvan Najarian. Big data analytics in healthcare. *BioMed research international*, 2015, 2015. 4

Duane Bender and Kamran Sartipi. HL7 FHIR: An agile and restful approach to healthcare information exchange. In *Proceedings of the 26th IEEE international symposium on computer-based medical systems*, pages 326–331. IEEE, 2013. 7

Tim Benson. *Principles of health interoperability HL7 and SNOMED*. Springer London, 2010. 9, 46

Shalini Bhartiya and Deepti Mehrotra. Challenges and recommendations to healthcare data exchange in an interoperable environment. *electronic Journal of Health Informatics*, 8(2):16, 2014. 5, 10

Saleh Binobaid, Ip-Shing Fan, and Mohammed Almeziny. Investigation interoperability problems in pharmacy automation: A case study in Saudi Arabia. 2016. 16

Alessio Botta, Walter De Donato, Valerio Persico, and Antonio Pescapé. Integration of cloud computing and internet of things: a survey. *Future generation computer systems*, 56:684–700, 2016. 4

Mark L Braunstein. *Health Informatics on FHIR: How HL7's New API is Transforming Healthcare*. Springer, 2018. 51

Bente Christensen and Gunnar Ellingsen. Evaluating model-driven development for large-scale ehrs through the openehr approach. *International journal of medical informatics*, 89:43–54, 2016. 26

Fernán González Bernaldo De Quirós, Carlos Otero, and Daniel Luna. Terminology services: Standard terminologies to control health vocabulary: Experience at the hospital italiano de buenos aires. *Yearbook of medical informatics*, 27(1):227, 2018. 48

Nicolas Delvaux, Bert Aertgeerts, Johan CH van Bussel, Geert Goderis, Bert Vaes, and Mieke Vermandere. Health data for research through a nationwide privacy-proof system in belgium: design and implementation. *JMIR medical informatics*, 6(4):e11428, 2018. 64

Hans Demski, Sebastian Garde, and Claudia Hildebrand. Open data models for smart health interconnected applications: the example of openehr. *BMC medical informatics and decision making*, 16(1):137, 2016. 15

Crescenzo Diomaiuta, Mario Sicuranza, Mario Ciampi, and Giuseppe De Pietro. A FHIR-based system for the generation and retrieval of clinical documents. In *ICT4AgeingWell*, pages 135–142, 2017. 12, 17

Asuman Dogac, Tuncay Namli, Alper Okcan, Gokce B. Laleci, and Marco Eichelberg. Key Issues of Technical Interoperability Solutions in eHealth and the RIDE. *ResearchGate*, Jan 2007. URL https://www.researchgate.net/publication/252378688_Key_Issues_of_Technical_Interoperability_Solutions_in_eHealth_and_the_RIDE. 6

Patricia C Dykes, Christine Caligtan, Andrew Novack, Debra Thomas, Linda Winfield, Gianna Zuccotti, and Roberto A Rocha. Development of automated quality reporting: aligning local efforts with national standards. In *AMIA Annual Symposium Proceedings*, volume 2010, page 187. American Medical Informatics Association, 2010. 21

David P Faxon and Anne Burgess. Cardiovascular registries: too much of good thing?, 2016. 38

Shelli L Feder. Data quality in electronic health records research: quality domains and assessment methods. *Western journal of nursing research*, 40(5):753–766, 2018. 41

Roy T Fielding and Richard N Taylor. *Architectural styles and the design of network-based software architectures*, volume 7. University of California, Irvine Irvine, 2000. 9

Ana Sofia Figueiredo. Data sharing: convert challenges into opportunities. *Frontiers in public health*, 5:327, 2017. 27

M Rachel Flynn, Conor Barrett, Francisco G Cosío, Anselm K Gitt, Lars Wallentin, Peter Kearney, Moira Lonergan, Emer Shelley, and Maarten L Simoons. The cardiology audit and registration data standards (cards), european data standards for clinical cardiology practice. *European heart journal*, 26(3):308–313, 2005. 38

Allan Fong, A Zachary Hettinger, and Raj M Ratwani. Exploring methods for identifying related patient safety events using structured and unstructured data. *Journal of biomedical informatics*, 58:89–95, 2015. 10

P Galetsi, K Katsaliaki, and S Kumar. Values, challenges and future directions of big data analytics in healthcare: A systematic review. *Social science & medicine*, 241:112533, 2019. 38

Goce Gavrilov, Elena Vlahu-Gjorgievska, and Vladimir Trajkovik. Healthcare data warehouse system supporting cross-border interoperability. *Health informatics journal*, 26 (2):1321–1332, 2020. 33

Kirstine Rosenbeck Gøeg, Rune Kongsgaard Rasmussen, Lasse Jensen, Christian Møller Wollesen, Søren Larsen, and Louise Bilenberg Pape-Haugaard. A future-proof architecture for telemedicine using loose-coupled modules and hl7 fhir. *Computer methods and programs in biomedicine*, 160:95–101, 2018. 53

Gayatri Gopal, Clemens Suter-Crazzolara, Luca Toldo, and Werner Eberhardt. Digital transformation in healthcare–architectures of present and future information technologies. *Clinical Chemistry and Laboratory Medicine (CCLM)*, 57(3):328–335, 2019. 13

Steven Grijzenhout and Maarten Marx. The quality of the xml web. *Journal of web semantics*, 19:59–68, 2013. 43

Christian Gulden, Romina Blasini, Azadeh Nassirian, Alexandra Stein, Fatma Betül Altun, Melanie Kirchner, Hans-Ulrich Prokosch, and Martin Boeker. Prototypical clinical trial registry based on fast healthcare interoperability resources (fhir): Design and implementation study. *JMIR Medical Informatics*, 9(1):e20470, 2021. 13

Birger Haarbrandt, Erik Tute, and Michael Marschollek. Automated population of an i2b2 clinical data warehouse from an openEHR-based data repository. *Journal of biomedical informatics*, 63:277–294, 2016. 15

Gregor Hohpe and Bobby Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004. 43

Taku Inohara, Shun Kohsaka, John A Spertus, Frederick A Masoudi, John S Rumsfeld, Kevin F Kennedy, Tracy Y Wang, Kyohei Yamaji, Tetsuya Amano, and Masato Nakamura. Comparative trends in percutaneous coronary intervention in japan and the united states, 2013 to 2017. *Journal of the American College of Cardiology*, 76(11):1328–1340, 2020. 27

Saadia Ismail, Majed Alshmari, Usman Qamar, Wasi Haider Butt, Khalid Latif, and Hafiz Farooq Ahmad. HL7 FHIR compliant data access model for maternal health information system. In *2016 IEEE 16th International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 51–56. IEEE, 2016. 9

Kipp W Johnson, Jessica Torres Soto, Benjamin S Glicksberg, Khader Shameer, Riccardo Miotto, Mohsin Ali, Euan Ashley, and Joel T Dudley. Artificial intelligence in cardiology. *Journal of the American College of Cardiology*, 71(23):2668–2679, 2018. 27

Gulraiz Javaid Joyia, Muhammad Usman Akram, Chaudary Naeem Akbar, and Muhammad Furqan Maqsood. Evolution of health level-7: A survey. In *Proceedings of the 2018 International Conference on Software Engineering and Information Management*, pages 118–123, 2018. 8

Ho-Won Jung, Seung-Gweon Kim, and Chang-Shin Chung. Measuring software product quality: A survey of iso/iec 9126. *IEEE software*, 21(5):88–92, 2004. 22

Mahtab Karami, Azin Rahimi, and Ali Hosseini Shahmirzadi. Clinical data warehouse: an effective tool to create intelligence in disease management. *The health care manager*, 36 (4):380–384, 2017. 31

Wajahat Ali Khan, Maqbool Hussain, Asad Masood Khattak, Muhammad Bilal Amin, and Sungyoung Lee. Saas based interoperability service for semantic mappings among health-care standards. In *8th International Conference on Innovations in Information Technology*, 2012. 29

RAMM Kieft, Erna M Vreeke, EM De Groot, PA Volkert, Anneke L Francke, and Diana MJ Delnoij. The development of a nursing subset of patient problems to support interoperability. *BMC medical informatics and decision making*, 17(1):1–12, 2017. 13

Athanasios Kiourtis, Argyro Mavrogiorgou, Andreas Menychtas, Ilias Maglogiannis, and Dimosthenis Kyriazis. Structurally mapping healthcare data to hl7 fhir through ontology alignment. *Journal of medical systems*, 43(3):62, 2019. 70

Clemens Scott Kruse, Rishi Goswamy, Yesha Jayendrakumar Raval, and Sarah Marawi. Challenges and opportunities of big data in health care: a systematic review. *JMIR medical informatics*, 4(4):e38, 2016. 10, 31

Yun Sik Kwak. International standards for building electronic health record (ehr). In *Proceedings of 7th International Workshop on Enterprise networking and Computing in Healthcare Industry, 2005. HEALTHCOM 2005.*, pages 18–23. IEEE, 2005. 9

Moritz Lehne, Julian Sass, Andrea Essenwanger, Josef Schepers, and Sylvia Thun. Why digital medicine depends on interoperability. *NPJ Digital Medicine*, 2(1):1–5, 2019. 5, 6, 10

Bei Li, Jianbin Li, Xiaoyun Lan, Ying An, Wuqiang Gao, and Yuqiao Jiang. Experiences of building a medical data acquisition system based on two-level modeling. *International journal of medical informatics*, 112:114–122, 2018. 15

Ehtisham Mahmud, Jesse Naghi, Lawrence Ang, Jonathan Harrison, Omid Behnamfar, Ali Pourdjabbar, Ryan Reeves, and Mitul Patel. Demonstration of the safety and feasibility of robotically assisted percutaneous coronary intervention in complex coronary lesions: results of the cora-pci study (complex robotically assisted percutaneous coronary intervention). *JACC: Cardiovascular Interventions*, 10(13):1320–1327, 2017. 66

Joshua C Mandel, David A Kreda, Kenneth D Mandl, Isaac S Kohane, and Rachel B Ramoni. Smart on FHIR: a standards-based, interoperable apps platform for electronic health records. *Journal of the American Medical Informatics Association*, 23(5):899–908, 2016. 9, 53

Agile Manifesto. Agile manifesto. *Haettu*, 14:2012, 2001. 9

Luis Marco-Ruiz, David Moner, José A Maldonado, Nils Kolstrup, and Johan G Bellika. Archetype-based data warehouse environment to enable the reuse of electronic health record data. *International journal of medical informatics*, 84(9):702–714, 2015. 15

Sebastian Mate, Felix Köpcke, Dennis Toddenroth, Marcus Martin, Hans-Ulrich Prokosch, Thomas Bürkle, and Thomas Ganslandt. Ontology-based data integration between clinical and research systems. *PloS one*, 10(1):e0116656, 2015. 36

Ahmed Mateen, Samina Kausar, and Ahsan Raza Sattar. A software reuse approach and its effect on software quality, an empirical study for the software industry. *arXiv preprint arXiv:1702.00125*, 2017. 21

Tobias A Mattei. Privacy, confidentiality, and security of health care information: lessons from the recent wannacry cyberattack. *World neurosurgery*, 104:972–974, 2017. 48

Alison J McMillan, Norman Swindells, Edward Archer, Alistair McIlhagger, Anna Sung, Kelvin Leong, and Rhys Jones. A review of composite product data interoperability and product life-cycle management challenges in the composites industry. *Advanced Manufacturing: Polymer & Composites Science*, 3(4):130–147, 2017. 6

Alejandro Metke-Jimenez, Jim Steel, David Hansen, and Michael Lawley. Ontoserver: a syndicated terminology server. *Journal of biomedical semantics*, 9(1):24, 2018. 11, 12, 17, 24, 48, 62

Lingtong Min, Qi Tian, Xudong Lu, Jiye An, and Huilong Duan. An openehr based approach to improve the semantic interoperability of clinical data registry. *BMC medical informatics and decision making*, 18(1):15, 2018. 15

Praveen Naik and Shantaram Nayak. Insights on research techniques towards cost estimation in software design. *International Journal of Electrical and Computer Engineering*, 7 (5):2883, 2017. 60

Lemai Nguyen, Emilia Bellucci, and Linh Thuy Nguyen. Electronic health records implementation: an evaluation of information system impact and contingency factors. *International journal of medical informatics*, 83(11):779–796, 2014. 28

Gibril J Njie, Krista K Proia, Anilkrishna B Thota, Ramona KC Finnie, David P Hopkins, Starr M Banks, David B Callahan, Nicolaas P Pronk, Kimberly J Rask, Daniel T Lackland, et al. Clinical decision support systems and prevention: a community guide cardiovascular disease systematic review. *American journal of preventive medicine*, 49(5):784–795, 2015. 27

Toan C. Ong, Michael G. Kahn, Bethany M. Kwan, Traci Yamashita, Elias Brandt, Patrick Hosokawa, Chris Uhrich, and Lisa M. Schilling. Dynamic-ETL: a hybrid approach for health data extraction, transformation and loading. *BMC Medical Informatics and Decision Making*, 17(1):1–12, Dec 2017. ISSN 1472-6947. doi: 10.1186/s12911-017-0532-3. 13, 14, 15, 17, 36

Vikas N O'Reilly-Shah, Katherine R Gentry, Wil Van Cleve, Samir M Kendale, Craig S Jabaley, and Dustin R Long. The covid-19 pandemic highlights shortcomings in us health care informatics infrastructure: a call to action. *Anesthesia and analgesia*, 2020. 5

Daniel Pascot, Faouzi Bouslama, and Sehl Mellouli. Architecturing large integrated complex information systems: an application to healthcare. *Knowledge and Information Systems*, 27(1):115–140, 2011. 13

Rune Pedersen, G Conceição, and Luis Marco Ruiz. Implementation of openehr in combination with clinical terminologies: Experiences from Norway. *Int J Adv Life Sciences (IJALS)*, 9:82–91, 2017. 16

Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007. 4, 22

Gordana M Radivojević, Gorana R Šormaz, and Bratislav S Lazić. Bi applications in logistics. In *1st Logistics International Conference Belgrade, Serbia*, pages 28–30, 2013. 15

Miriam Reisman. Ehrs: the challenge of making electronic data usable and interoperable. *Pharmacy and Therapeutics*, 42(9):572, 2017. 34

Albrecht Richen and Ansgar Steinhorst. Standardization or harmonization? you need both. *European Health Informatics (November 2005)*, 5, 2005. 15

Rishi Saripalle, Mehdi Sookhak, and Mahboobeh Haghparast. An interoperable umls terminology service using fhir. *Future Internet*, 12(11):199, 2020. 11

Douglas C Schmidt. Model-driven engineering. *Computer-IEEE Computer Society-*, 39(2): 25, 2006. 15

Todd Sedano. Code readability testing, an empirical study. In *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*, pages 111–117. IEEE, 2016. 41

Ilia Semenov, Georgy Kopanitsa, Dmitry Denisov, Yakovenko Alexandr, Roman Osenev, and Yury Andreychuk. Patients decision aid system based on fhir profiles. *Journal of medical systems*, 42(9):166, 2018. 13

Alkis Simitsis, Panos Vassiliadis, Umeshwar Dayal, Anastasios Karagiannis, and Vasiliki Tziovara. Benchmarking etl workflows. In *Technology Conference on Performance Evaluation and Benchmarking*, pages 199–220. Springer, 2009. 36

Joyce Simons. Registratie aan de bron–eenduidig en eenmalig vastleggen voor hergebruik. In *De dokter en digitalisering*, pages 71–79. Springer, 2019. 13

Steven R Steinhubl and Eric J Topol. Moving from digitalization to digitization in cardiovascular care: why is it important, and what could it mean for patients and providers? *Journal of the American College of Cardiology*, 66(13):1489–1496, 2015. 27

Vasileios Theodorou, Petar Jovanovic, Alberto Abellò, and Emona Nakuçi. Data generator for evaluating etl process quality. *Information Systems*, 63:80–100, 2017. 22

Michael A Tutty, Lindsey E Carlasare, Stacy Lloyd, and Christine A Sinsky. The complex case of ehrs: examining the factors impacting the ehr user experience. *Journal of the American Medical Informatics Association*, 26(7):673–677, 2019. 31

Loan R Van Hoeven, Martine C De Bruijne, Peter F Kemper, Maria MW Koopman, Jan MM Rondeel, Anja Leyte, Hendrik Koffijberg, Mart P Janssen, and Kit CB Roes. Validation of multisource electronic health record data: an application to blood transfusion data. *BMC medical informatics and decision making*, 17(1):107, 2017. 41

Jonathan Waring, Charlotta Lindvall, and Renato Umeton. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artificial Intelligence in Medicine*, 104:101822, 2020. 5

Luis Weir. *Enterprise API Management: Design and deliver valuable business APIs*. Packt Publishing Ltd, 2019. 13

APPENDIX

```
<ResultField>
  <Name Value="Patient.Voornaam" />
  <Title Value="First_Name2010" />
  <ColumnWidth Value="40" />
  <Level Value="0" />
  <Column Value="-1" />
  <ColumnSpan Value="1" />
  <Row Value="-1" />
  <RowSpan Value="1" />
  <UseStarWidth Value="False" />
  <UseFirstLineOnly Value="False" />
  <StarUnits Value="1" />
  <MinWidth Value="60" />
  <UseTextWrapping Value="False" />
  <ActionLinkStream Value="" />
</ResultField>
```

Figure 40: Example of ChipSoft extraction query where data is directly accessed

```
<ResultField>
  <Name
  Value="=ValueAsString('Gegevens.PatiëntkarakteristiekenCS00396300.RisicofactorenCS00357976.RokenCS00345581OBJ.
  Omschrijving')" />
  <Title Value="Tobacco_Use4625" />
  <ColumnWidth Value="40" />
  <Level Value="0" />
  <Column Value="-1" />
  <ColumnSpan Value="1" />
  <Row Value="-1" />
  <RowSpan Value="1" />
  <UseStarWidth Value="False" />
  <UseFirstLineOnly Value="False" />
  <StarUnits Value="1" />
  <MinWidth Value="60" />
  <UseTextWrapping Value="False" />
  <ActionLinkStream Value="" />
</ResultField>
```

Figure 41: Example of ChipSoft extraction query where data is accessed through a simple expression

```
<ResultField>
  <Name Value="=setVar('varUitgevoerdeTesten',
  ValueAsCollection('Gegevens.IndicatiesEnTijdenCS00396307.UitgevoerdeTestenCS00396378COLL'));
  {if(Count(FilterColl(varUitgevoerdeTesten,'Omschrijving','electrocardiogram bij rust','electrocardiogram met
  stress-test')) &gt; 0,'ECG',if((Count(FilterColl(varUitgevoerdeTesten,'Omschrijving','geen test (staged
  procedure)')) = 1) and Count(varUitgevoerdeTesten) = 1,'None',''))};
  mapconc(varUitgevoerdeTesten,Expr(`Display('Omschrijving')`),'|')" />
  <Title Value="Electrocardiac_Assessment_Method5037bis" />
  <ColumnWidth Value="40" />
  <Level Value="0" />
  <Column Value="-1" />
  <ColumnSpan Value="1" />
  <Row Value="-1" />
  <RowSpan Value="1" />
  <UseStarWidth Value="False" />
  <UseFirstLineOnly Value="False" />
  <StarUnits Value="1" />
  <MinWidth Value="60" />
  <UseTextWrapping Value="False" />
  <ActionLinkStream Value="" />
</ResultField>
```

Figure 42: Example of ChipSoft extraction query where data is accessed through applying a more advanced expression

FHIR resource	***resource content***
Organization	A representation of a medical institution where the cardiovascular study is performed.
Patient	Demographics and other administrative information about an individual undergoing a cardiovascular examination.
Observation	A representation of a measurement and simple assertions made about a patient, device or other subject.
Encounter	An interaction between a patient and healthcare provider for the purpose of providing healthcare service(s) or assessing the health status of a patient.
Practitioner	A person who is directly or indirectly involved in the provisioning of healthcare.
FamilyMemberHistory	Significant health events and conditions for a person related to the patient relevant in the context of care for the patient.
Provenance	Provenance of a resource is a record that describes entities and processes involved in producing and delivering or otherwise influencing that resource.

Table 1: Description of used FHIR resources http://hl7.org/fhir/STU3/

```
map mainChipSoft( <- Input::ChipSoft in, -> Output::feed out )
{
    //map FHIR Organization Resource
    idxEntry = sizeof(out.entry);
    mapOrganizationResource(in,out);

    //map FHIR Practitioner Resource
    idxEntry = sizeof(out.entry);
    mapPractitionerResource(in,out);

    //map FHIR Patient Resource
    idxEntry = sizeof(out.entry);
    mapPatientResource(in,out);

    //map FHIR Encounter Resource
    idxEntry = sizeof(out.entry);
    mapEncounterResource(in,out);

    //map FHIR Observation Resource
    idxEntry = sizeof(out.entry);
    for (int i = 0; i < sizeof(in.Extraction.Observations.Observation); i = i + 1)
    {

        if (!(isnull(in.Extraction.Observations.Observation[i].Value.#PCDATA)) && !(isempty(in.Extraction.Observations.Observation[i].Value.#PCDATA)))
        {
            idxEntry = sizeof(out.entry);
            observationID = i;
            RhapsodyTableLookup(gValueX,"IBE_ILS_Common_V1_Toolkit_FHIR_ObservationIdentifier_Lookup","ValueX","","Value_To_Be_Looked_Up",in.Extraction.Observations.Observation[i].Code.#PCDATA);
            if(!(gValueX==""))
                mapObservationResource(in,out,in.Extraction.Observations.Observation[i].Code.#PCDATA,gValueX,in.Extraction.Observations.Observation[i].Value.#PCDATA);
        }
    }

    //map FHIR Procedure resource
    idxEntry = sizeof(out.entry);
    mapProcedureResource(in,out);


    //map FHIR FamilyMemberHistory resource
    idxEntry = sizeof(out.entry);
    if (!(isnull(in.Extraction.FamilyMemberHistory.Family_Hx_of_Premature_CAD.Value.#PCDATA)) && !(isempty(in.Extraction.FamilyMemberHistory.Family_Hx_of_Premature_CAD.Value.#PCDATA)))
        {
            idxEntry = sizeof(out.entry);
            //FamilyMemberHistory resource not available into the normalized schema. Mapping realized using the "Other" resource. Mapping to FamilyMemberHistory done in DSTU1 to DSTU3 mapper
            mapOther(in,out);
        }

}
```

Figure 43: Modularization of the ETL process

Parameter	Description
output	points to a global variable containing the lookup result if successful, or the default value id if no match is found in the lookup table.
tablename	contains the name of the lookup table in Rhapsody.
resultColumnName	references the column's name within the lookup table that should be used for the result.
defaultValue	represents the returned default value if no match is found.
queryColumn1	represents the column name to find a matching row.
queryValue1	contains the value to perform a lookup.
queryColumn2	represents an additional column name to find a matching row.
queryValue2	contains the value to perform a lookup.

Figure 44: Description of RhapsodyTableLookup() function parameters

```
{
 "resourceType": "Bundle",
 "type": "transaction",
 "entry": [
  {
   "fullUrl": "Organization/Hospital",
   "resource": {
    "resourceType": "Organization",
    "meta": {
     "profile": [
      "http://mygraduation.org/fhir/StructureDefinition/Organization"
     ]
    },
    "identifier": [
     {
      "value": "Hospital"
     }
    ],
    "name": "Test hospital"
   },
   "request": {
    "method": "PUT",
    "url": "Organization?identifier=Hospital"
   }
  },
  {
   "fullUrl": "Practitioner/123123321321PhysicianFirstName1PhysicianName1",
   "resource": {
    "resourceType": "Practitioner",
    "meta": {
     "profile": [
      "http://mygraduation.org/fhir/StructureDefinition/Practitioner"
     ]
    },
    "identifier": [
     {
      "use": "official",
      "system": "https://www.hospital.be/identifiers/PractitionerIdentifier",
      "value": "123123321321"
     }
    ],
    "name": [
     {
      "family": "Physician Name 1",
      "given": [
       "Physician FirstName 1"
      ]
     }
    ]
   },
   "request": {
    "method": "PUT",
    "url": "Practitioner?identifier=https://www.hospital.be/identifiers/PractitionerIdentifier|123123321321"
   }
  },
  {
   "fullUrl": "Patient/id1",
   "resource": {
    "resourceType": "Patient",
    "meta": {
     "profile": [
      "http://mygraduation.org/fhir/StructureDefinition/Patient"
     ]
    },
    "identifier": [
     {
      "type": {
       "coding": [
```

```
        {
         "system": "http://hl7.org/fhir/v2/0203",
         "code": "MR",
         "display": "Medical record number"
        }
       ]
      },
      "system": "https://www.hospital.be/identifiers/PatientIdentifier",
      "value": "id1"
     }
    ],
    "active": true,
    "name": [
     {
      "use": "usual",
      "family": "Pat1"
     }
    ],
    "gender": "female",
    "birthDate": "1946-02-08",
    "address": [
     {
      "postalCode": "8770",
      "country": "B"
     }
    ],
    "generalPractitioner": [
     {
      "reference": "Practitioner/123123321321PhysicianFirstName1PhysicianName1"
     }
    ],
    "managingOrganization": {
     "reference": "Organization/Hospital "
    }
   },
   "request": {
    "method": "PUT",
    "url": "Patient?identifier=https://www.hospital.be/identifiers/PatientIdentifier|id1"
   }
  },
  {
   "fullUrl": "Observation/CathlabID0032584579ObservationCode4615",
   "resource": {
    "resourceType": "Observation",
    "meta": {
     "profile": [
      "http://mygraduation.org/fhir/StructureDefinition/ObservationvalueBoolean"
     ]
    },
    "identifier": [
     {
      "value": "CathlabID0032584579_ObservationCode4615"
     }
    ],
    "status": "final",
    "code": {
     "coding": [
      {
       "system": "http://snomed.info/sct",
       "code": "38341003",
       "display": "Indicate if the patient has a current diagnosis of hypertension."
      }
     ]
    },
    "subject": {
     "reference": "Patient/id1"
    },
    "context": {
```

```
    "reference": "Encounter/Encid1"
   },
   "valueBoolean": true
  },
  "request": {
   "method": "PUT",
   "url": "Observation?identifier=CathlabID0032584579_ObservationCode4615"
  }
 },
 {
  "fullUrl": "Observation/CathlabID0032584579ObservationCode4011",
  "resource": {
   "resourceType": "Observation",
   "meta": {
    "profile": [
     "http://mygraduation.org/fhir/StructureDefinition/ObservationvalueCodeableConcepts"
    ]
   },
   "identifier": [
    {
     "value": "CathlabID0032584579_ObservationCode4011"
    }
   ],
   "status": "final",
   "code": {
    "coding": [
     {
      "system": "http://snomed.info/sct",
      "code": "420816009",
      "display": "Indicate the patient's latest dyspnea or functional class, coded as the New York Heart Association (NYHA) classification."
     }
    ]
   },
   "subject": {
    "reference": "Patient/id1"
   },
   "context": {
    "reference": "Encounter/Encid1"
   },
   "valueCodeableConcept": {
    "coding": [
     {
      "system": "http://snomed.info/sct",
      "code": "420913000",
      "display": "Class III"
     }
    ]
   }
  },
  "request": {
   "method": "PUT",
   "url": "Observation?identifier=CathlabID0032584579_ObservationCode4011"
  }
 },
 {
  "fullUrl": "Observation/CathlabID0032584579ObservationCode8505",
  "resource": {
   "resourceType": "Observation",
   "meta": {
    "profile": [
     "http://mygraduation.org/fhir/StructureDefinition/ObservationHemoglobin_8505"
    ]
   },
   "identifier": [
    {
     "value": "CathlabID0032584579_ObservationCode8505"
    }
   ],
```

```
  "status": "final",
  "code": {
   "coding": [
    {
     "system": "http://loinc.org",
     "code": "718-7",
     "display": "Indicate the hemoglobin (Hgb) value in g/dL."
    }
   ]
  },
  "subject": {
   "reference": "Patient/id1"
  },
  "context": {
   "reference": "Encounter/Encid1"
  },
  "effectiveDateTime": "2020-01-29T20:20:00+01:00",
  "valueQuantity": {
   "value": 12.9,
   "unit": "g/dL",
   "system": "http://unitsofmeasure.org",
   "code": "g/dL"
  }
 },
 "request": {
  "method": "PUT",
  "url": "Observation?identifier=CathlabID0032584579_ObservationCode8505"
 }
},
{
 "fullUrl": "Observation/CathlabID0032584579ObservationCode7500",
 "resource": {
  "resourceType": "Observation",
  "meta": {
   "profile": [
    "http://mygraduation.org/fhir/StructureDefinition/ObservationvalueCodeableConcepts"
   ]
  },
  "identifier": [
   {
    "value": "CathlabID0032584579_ObservationCode7500"
   }
  ],
  "status": "final",
  "code": {
   "coding": [
    {
     "system": "http://snomed.info/sct",
     "code": "253727002",
     "display": "Indicate the dominance of the coronary anatomy."
    }
   ]
  },
  "subject": {
   "reference": "Patient/id1"
  },
  "context": {
   "reference": "Encounter/Encid1"
  },
  "valueCodeableConcept": {
   "coding": [
    {
     "system": "http://snomed.info/sct",
     "code": "253728007",
     "display": "Right dominant coronary system"
    }
   ]
  }
```

```
 },
 "request": {
  "method": "PUT",
  "url": "Observation?identifier=CathlabID0032584579_ObservationCode7500"
 }
},
{
 "fullUrl": "Observation/CathlabID0032584579ObservationCode7825CS",
 "resource": {
  "resourceType": "Observation",
  "meta": {
   "profile": [
    "http://mygraduation.org/fhir/StructureDefinition/ObservationvalueCodeableConcepts"
   ]
  },
  "identifier": [
   {
    "value": "CathlabID0032584579_ObservationCode7825CS"
   }
  ],
  "status": "final",
  "code": {
   "coding": [
    {
     "system": "http://chipsoft.nl/standardcontent",
     "code": "CS00213355",
     "display": "Indicate the reason the percutaneous coronary intervention PCI is being performed."
    }
   ]
  },
  "subject": {
   "reference": "Patient/id1"
  },
  "context": {
   "reference": "Encounter/Encid1"
  },
  "valueCodeableConcept": {
   "coding": [
    {
     "system": "http://snomed.info/sct",
     "code": "194828000",
     "display": "Angina pectoris"
    },
    {
     "system": "http://snomed.info/sct",
     "code": "58158008",
     "display": "stabiel"
    }
   ]
  }
 },
 "request": {
  "method": "PUT",
  "url": "Observation?identifier=CathlabID0032584579_ObservationCode7825CS"
 }
},
{
 "fullUrl": "Procedure/Procid15464f05e90c64cc2abe585bd462cb0db",
 "resource": {
  "resourceType": "Procedure",
  "meta": {
   "profile": [
    "http://mygraduation.org/fhir/StructureDefinition/Procedure"
   ]
  },
  "identifier": [
   {
    "value": "PCI_0032584579"
```

```
    }
   ],
   "status": "completed",
   "code": {
    "coding": [
     {
      "system": "http://snomed.info/sct",
      "code": "415070008",
      "display": "Percutaneous coronary intervention"
     }
    ]
   },
   "subject": {
    "reference": "Patient/id1"
   },
   "context": {
    "reference": "Encounter/Encid1"
   },
   "performedPeriod": {
    "start": "2020-01-29T11:09:00+01:00",
    "end": "2020-01-29T12:14:00+01:00"
   },
   "performer": [
    {
     "actor": {
      "reference": "Practitioner/123123321321PhysicianFirstName1PhysicianName1"
     }
    }
   ],
   "bodySite": [
    {
     "coding": [
      {
       "system": "http://snomed.info/sct",
       "code": "368503001",
       "display": "art. radialis rechts"
      }
     ]
    }
   ]
  },
  "request": {
   "method": "PUT",
   "url": "Procedure?identifier=PCI_0032584579"
  }
 },
 {
  "fullUrl": "FamilyMemberHistory/FMH0032584579",
  "resource": {
   "resourceType": "FamilyMemberHistory",
   "meta": {
    "profile": [
     "http://mygraduation.org/fhir/StructureDefinition/FamilyMemberHistory"
    ]
   },
   "identifier": [
    {
     "value": "FMH_0032584579"
    }
   ],
   "status": "completed",
   "patient": {
    "reference": "Patient/id1"
   },
   "relationship": {
    "text": "no relationship available"
   },
   "condition": [
```

```
      {
        "code": {
          "coding": [
            {
              "system": "http://snomed.info/sct",
              "code": "373067005",
              "display": "No"
            }
          ]
        }
      }
    ]
  },
  "request": {
    "method": "PUT",
    "url": "FamilyMemberHistory?identifier=FMH_0032584579"
  }
 }
]
}
```


Figure 45: Partial example of anonymized FHIR output generated by the prototype.

```
<StructureDefinition
  xmlns="http://hl7.org/fhir">
  <url value="http://mygraduation.org/fhir/StructureDefinition/HeightQuantity"/>
  <name value="HeightQuantity"/>
  <status value="draft"/>
  <fhirVersion value="3.0.2"/>
  <mapping>
    <identity value="v2"/>
    <uri value="http://hl7.org/v2"/>
    <name value="HL7 v2 Mapping"/>
  </mapping>
  <mapping>
    <identity value="rim"/>
    <uri value="http://hl7.org/v3"/>
    <name value="RIM Mapping"/>
  </mapping>
  <kind value="complex-type"/>
  <abstract value="false"/>
  <type value="Quantity"/>
  <baseDefinition value="http://hl7.org/fhir/StructureDefinition/Quantity"/>
  <derivation value="constraint"/>
  <differential>
    <element id="Quantity.value">
      <path value="Quantity.value"/>
      <min value="1"/>
      <minValueDecimal value="0.0"/>
      <maxValueDecimal value="250.0"/>
    </element>
    <element id="Quantity.system">
      <path value="Quantity.system"/>
      <min value="1"/>
      <fixedUri value="http://unitsofmeasure.org"/>
    </element>
    <element id="Quantity.code">
      <path value="Quantity.code"/>
      <min value="1"/>
      <fixedCode value="cm"/>
    </element>
  </differential>
</StructureDefinition>
```

Figure 46: Example of a FHIR Structure Definition restricting Height Quantity datatype